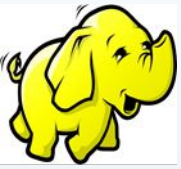



hadoop

History of Hadoop

Doug Cutting

Reads paper

"GFS and Map-reduce" 2004

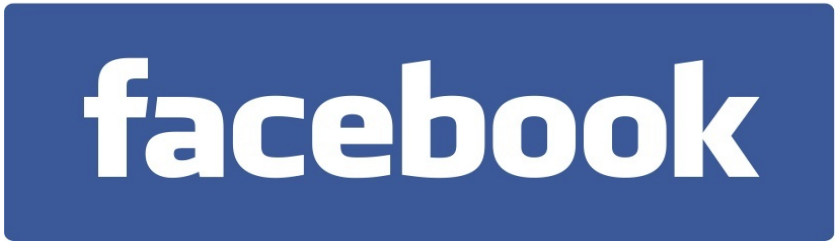
"It is an important technique!"

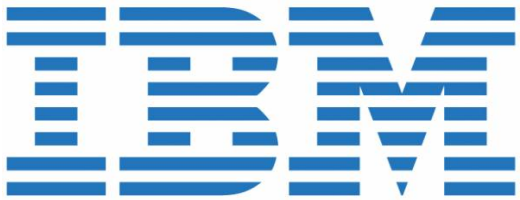
Joins Yahoo! at 2006

Apache Nutch

Extended

Who uses Hadoop?





Hadoop offers

- Redundant, Fault-tolerant data storage
- Parallel computation framework
- Job coordination

Programmers

No longer need to worry about

Q: Where file is located?

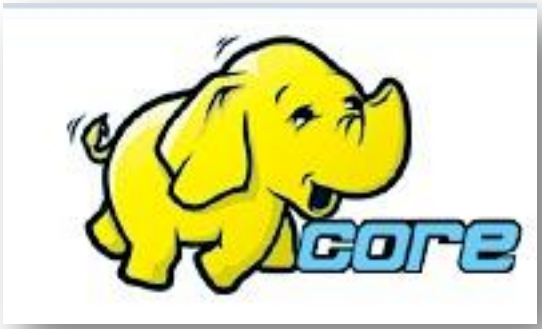
Q: How to handle failures & data lost?

Q: How to divide computation?

Q: How to program for scaling?

Common Hadoop Distributions

- Open Source
  - Apache



- Commercial
  - Cloudera
  - Hortonworks
  - MapR
  - AWS MapReduce
  - Microsoft Azure HDInsight (Beta)

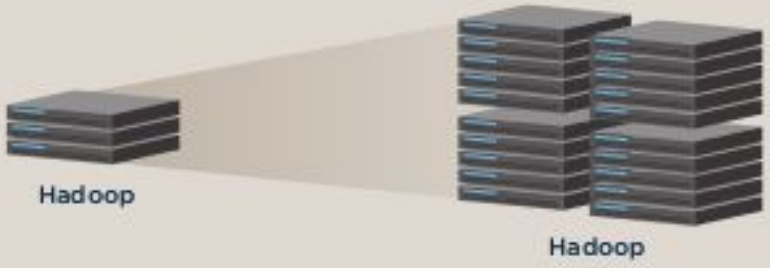
Why Use Hadoop?

Hadoop provides 4 key breakthroughs compared to traditional solutions:

1 Overcomes the traditional limitations of storage and compute.

2 Leverage inexpensive, commodity hardware as the platform.

3 Provides linear scalability from 1 to 4000 servers.



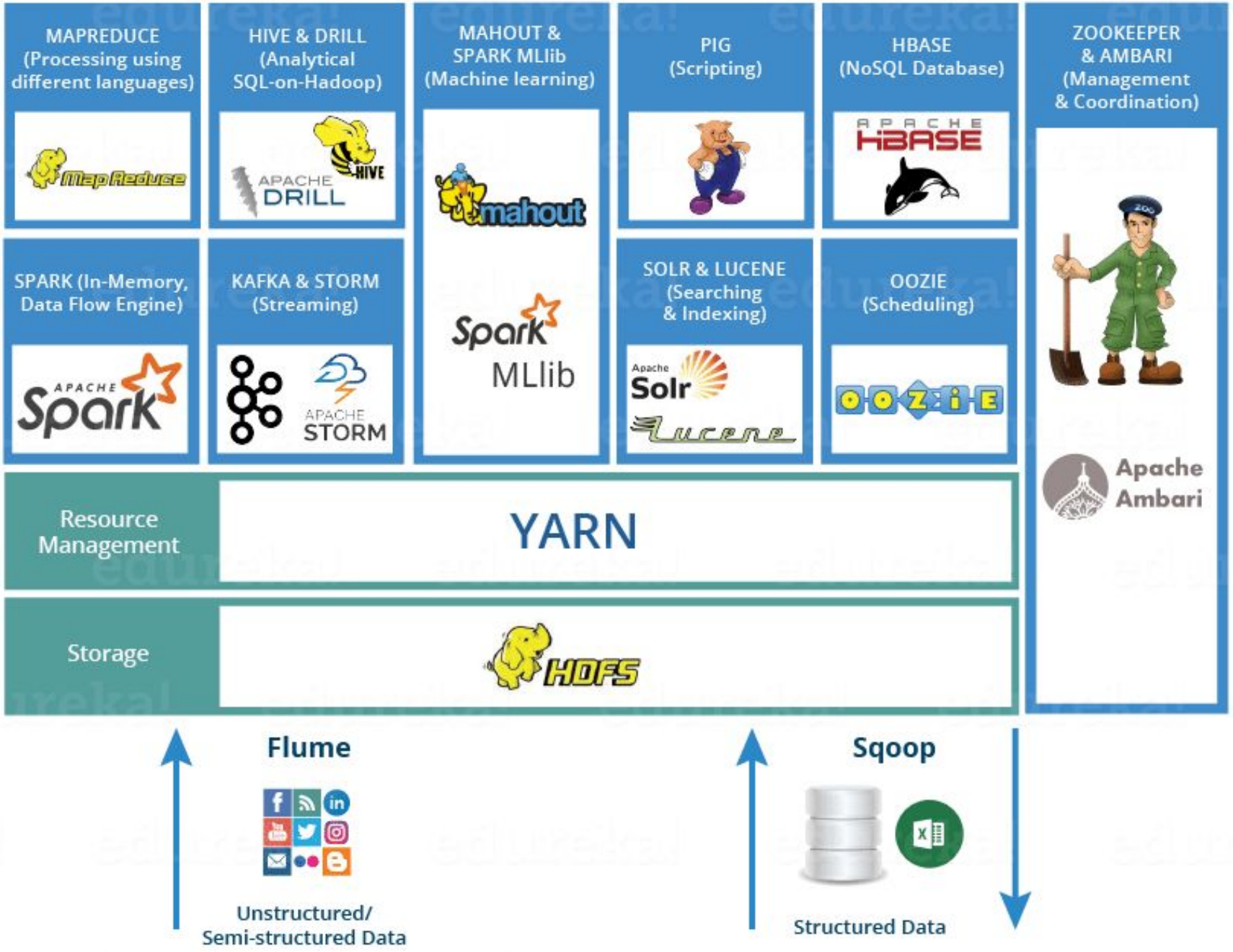
4 Low cost, open source software.

Where is Hadoop used?

Industry	Use Cases
Technology	Search People you may know Movie recommendations
Banks	Fraud Detection Regulatory Risk management
Media Retail	Marketing analytics Customer service Product recommendations
Manufacturing	Preventive maintenance

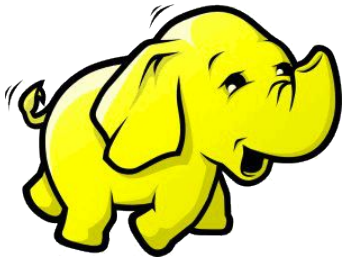
Companies Using Hadoop

Hadoop Ecosystem

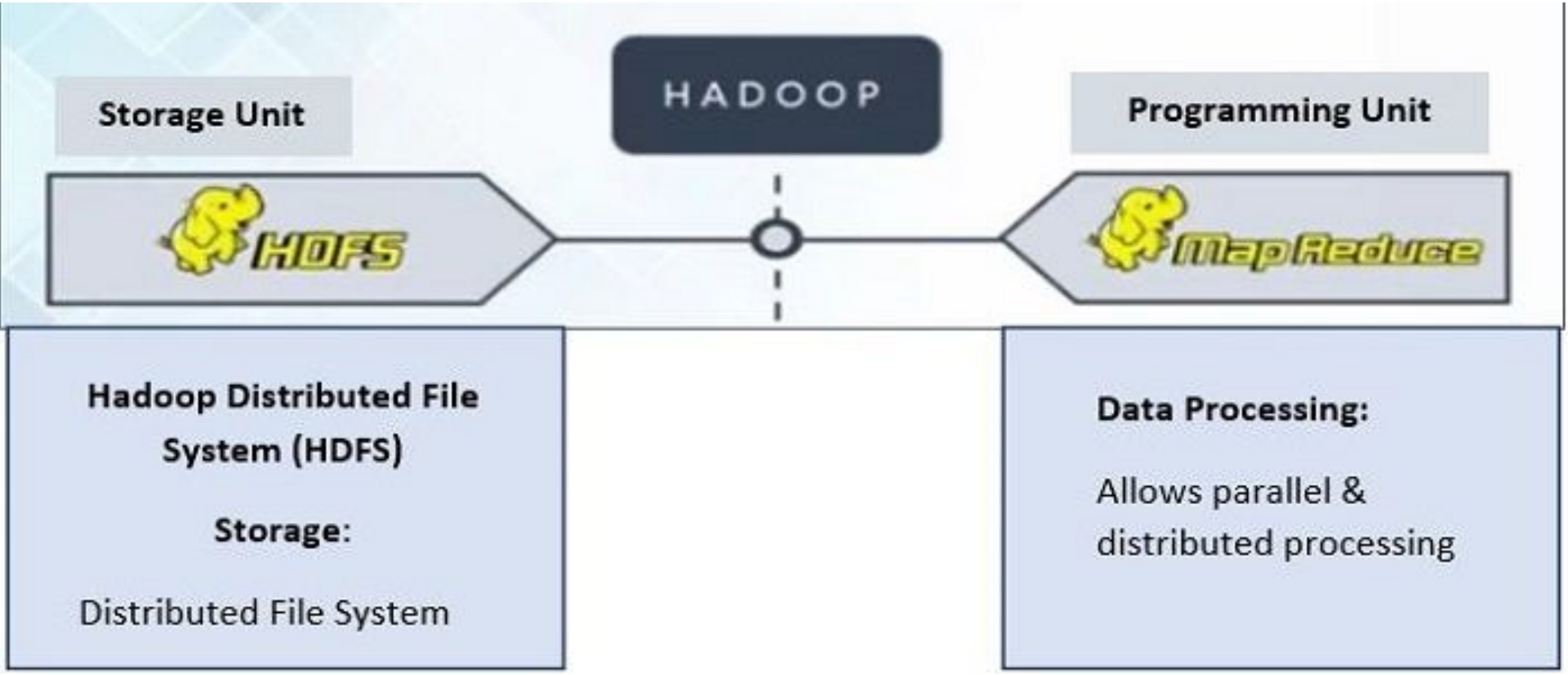


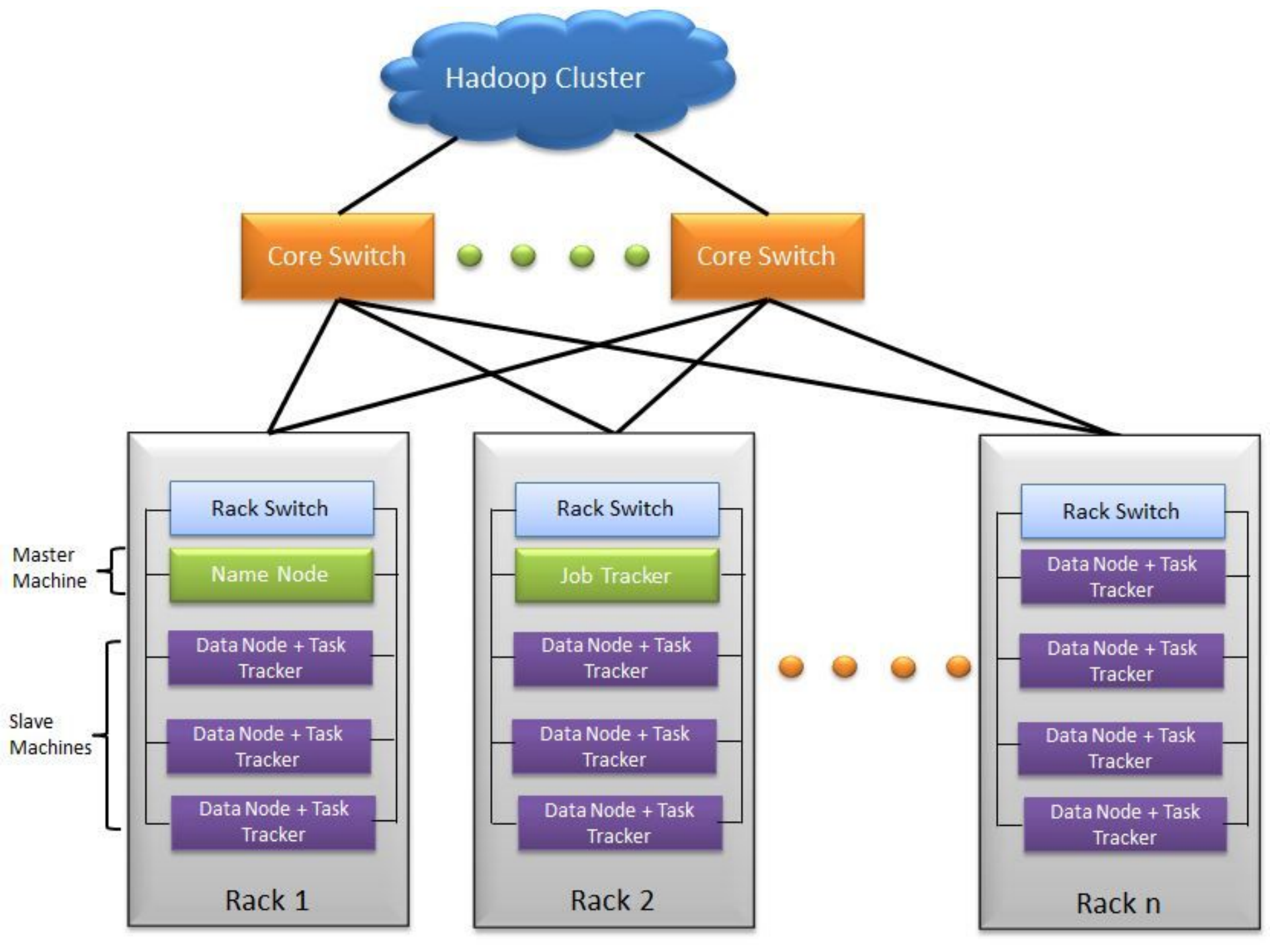
Comparing: RDBMS vs. Hadoop

	Traditional RDBMS	Hadoop / MapReduce
Data Size	Gigabytes (Terabytes)	Petabytes (Hexabytes)
Access	Interactive and Batch	Batch – NOT Interactive
Updates	Read / Write many times	Write once, Read many times
Structure	Static Schema	Dynamic Schema
Integrity	High (ACID)	Low
Scaling	Nonlinear	Linear
Query Response Time	Can be near immediate	Has latency (due to batch processing)



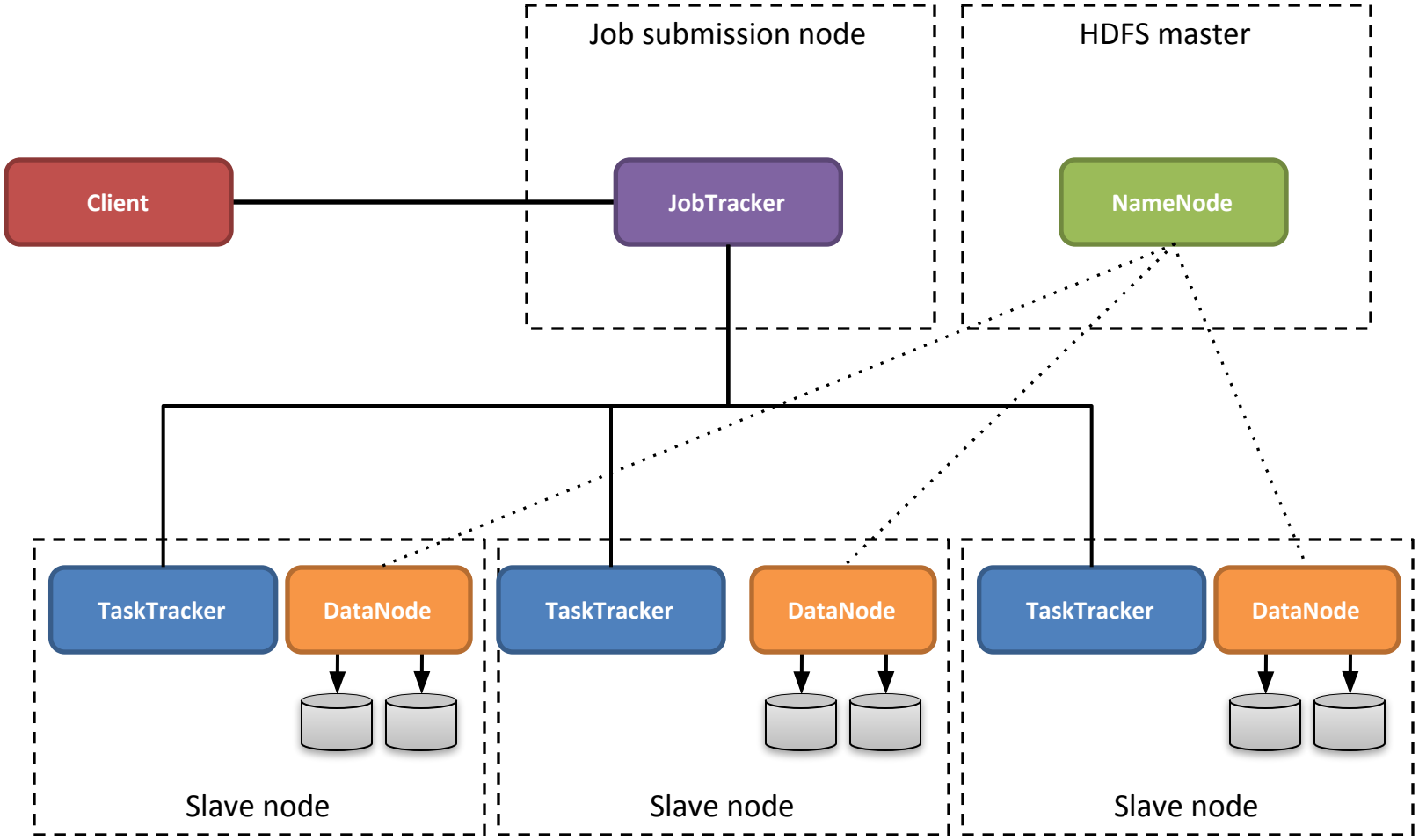
Basic Components of Hadoop




Hadoop Cluster
Core Switch
Core Switch
Rack Switch
Name Node
Data Node + Task Tracker
Data Node + Task Tracker
Data Node + Task Tracker
Master Machine
Slave Machines
Rack 1
Rack Switch
Job Tracker
Data Node + Task Tracker
Data Node + Task Tracker
Data Node + Task Tracker
Rack 2
Rack Switch
Data Node + Task Tracker
Data Node + Task Tracker
Data Node + Task Tracker
Data Node + Task Tracker
Rack n

Typical Hadoop Cluster

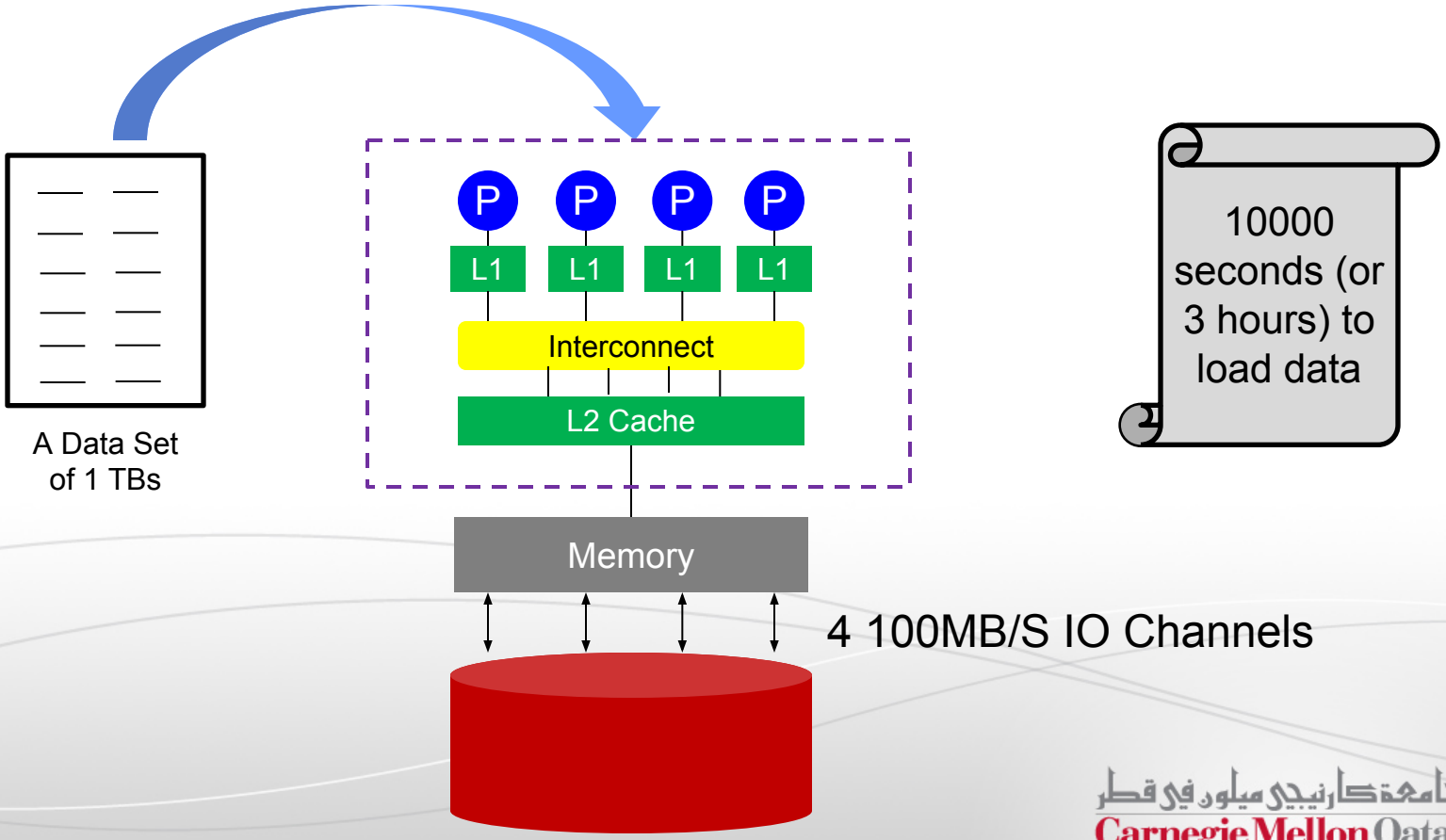
Hadoop Cluster Architecture



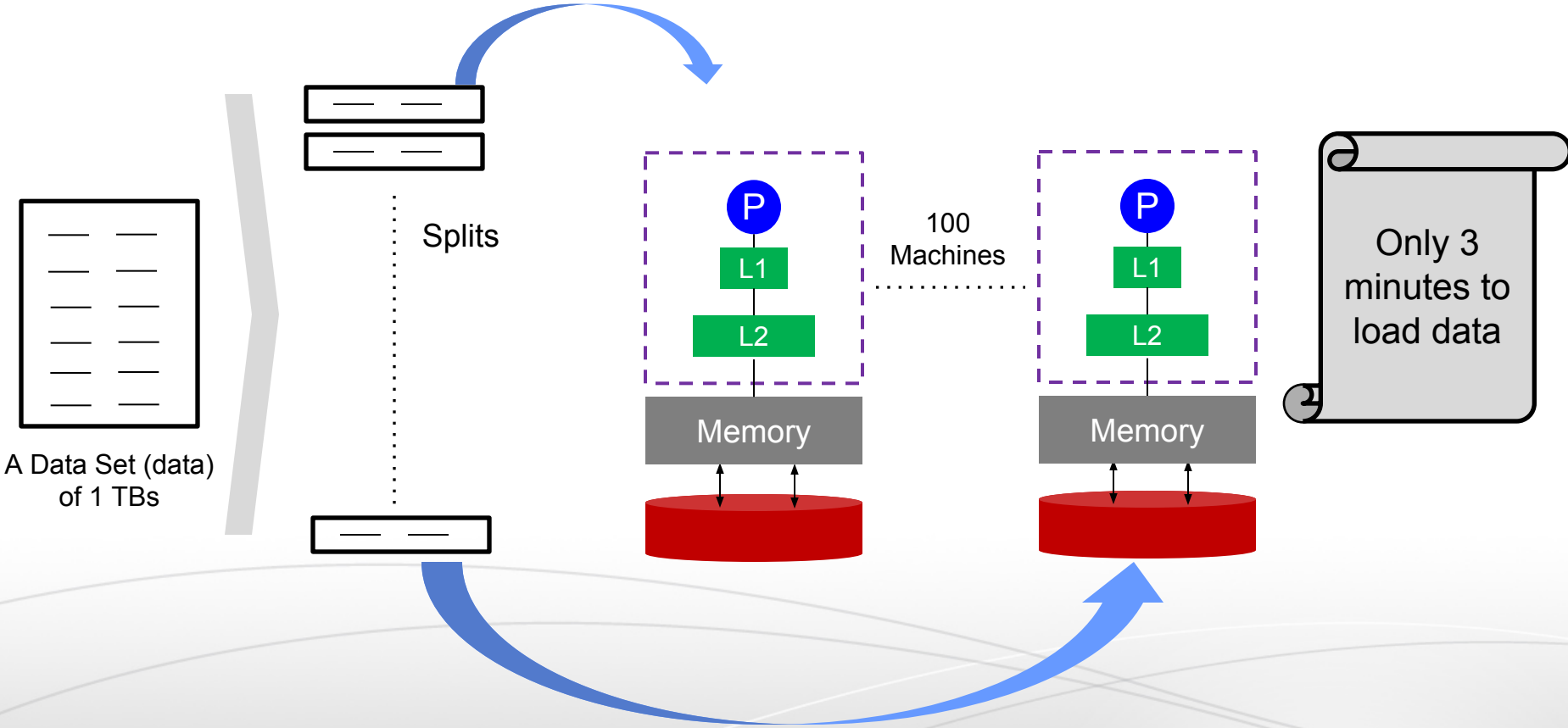
Hadoop Distributed File System

Why HDFS?

- Even if 100s or 1000s of cores are placed on a CMP, it is a challenge to deliver input data to these cores fast enough for processing.



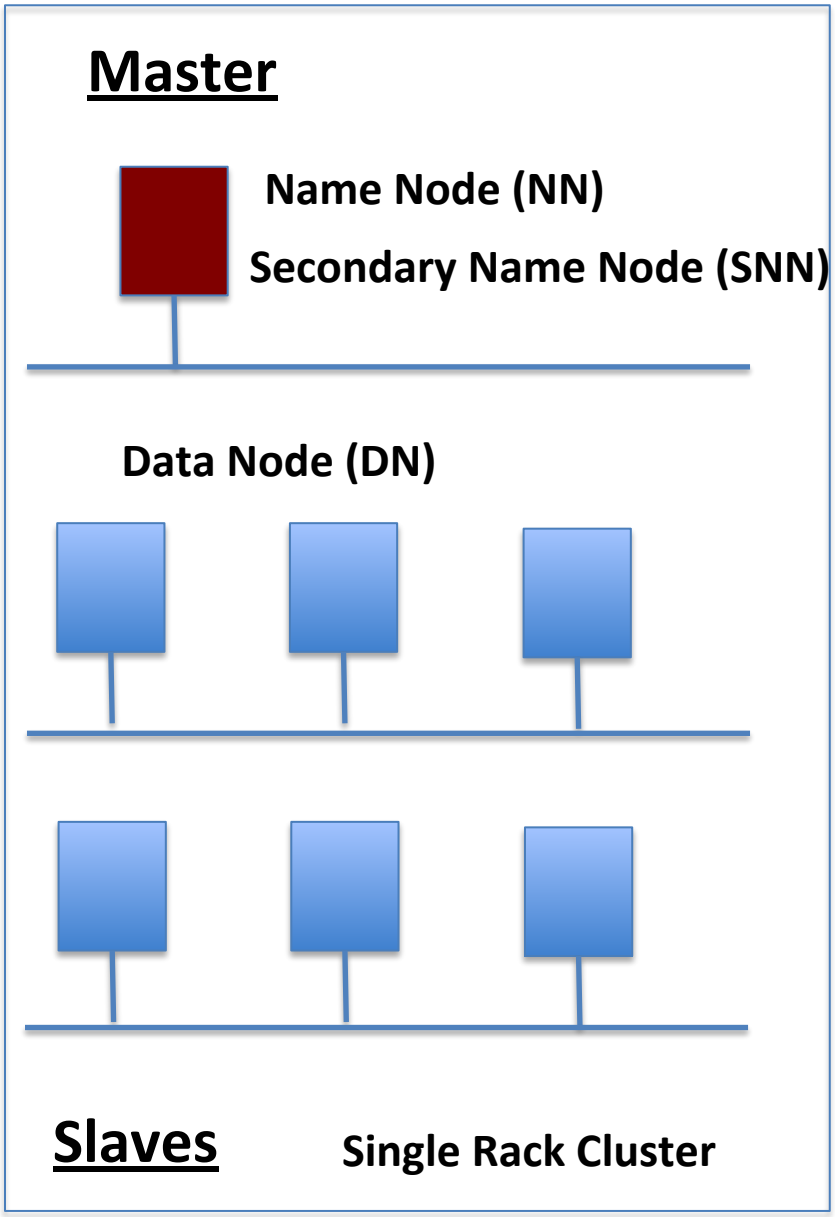
Why HDFS?



HDFS Features

- Large data sets and files
  - Support **Petabytes** size
- Heterogeneous
  - Could be deployed on **different hardware**
- Streaming data access
  - **Batch** processing rather than interactive user access
- Fault-Tolerance
  - Automatic recovery or report failure

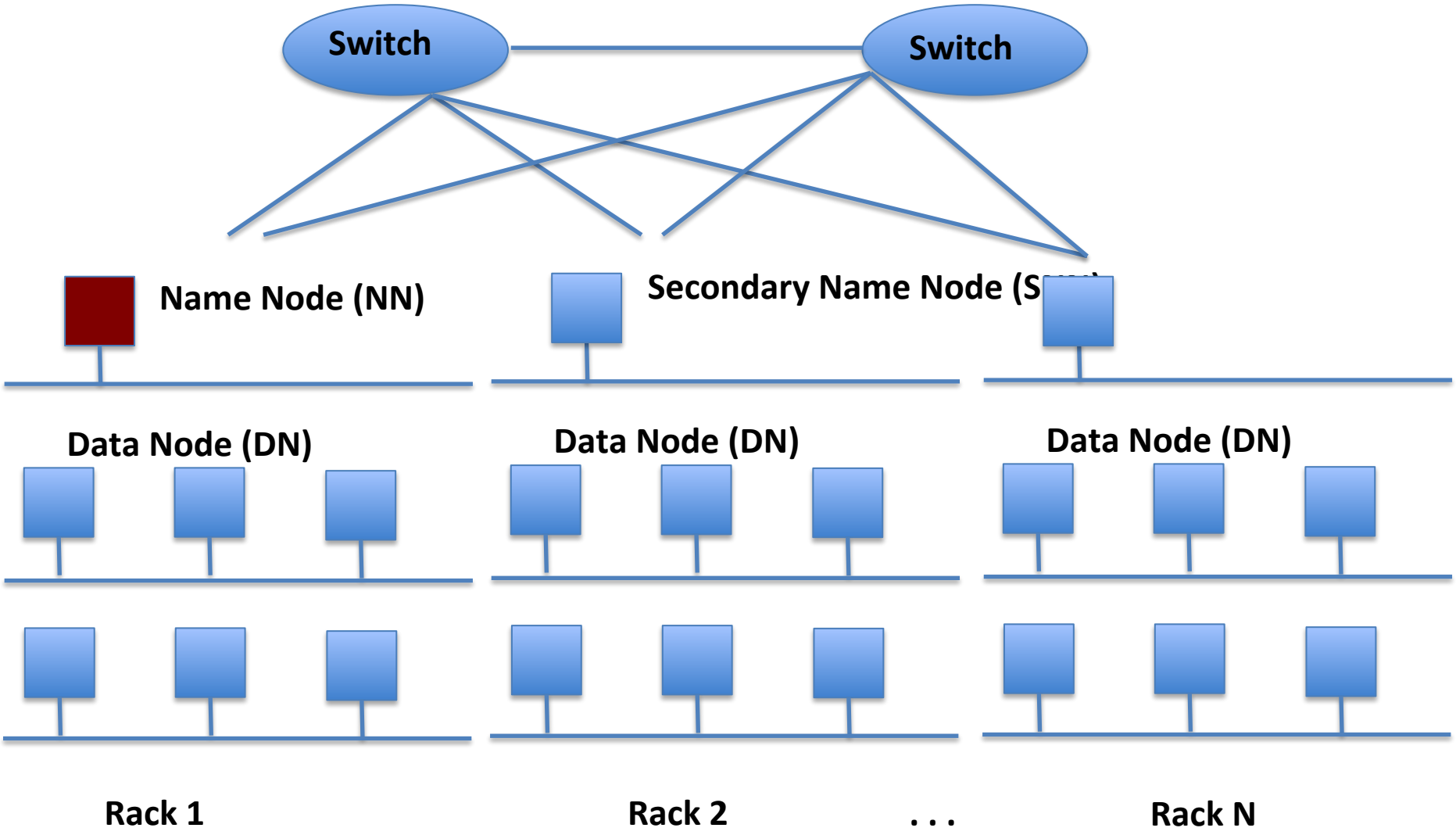
HDFS Architecture: Master-Slave



- Name Node: Controller
  - File System Name Space Management
  - Block Mappings
- Data Node: Work Horses
  - Block Operations
  - Replication
- Secondary Name Node:
  - Checkpoint node

HDFS Architecture: Master-Slave

Multiple-Rack Cluster

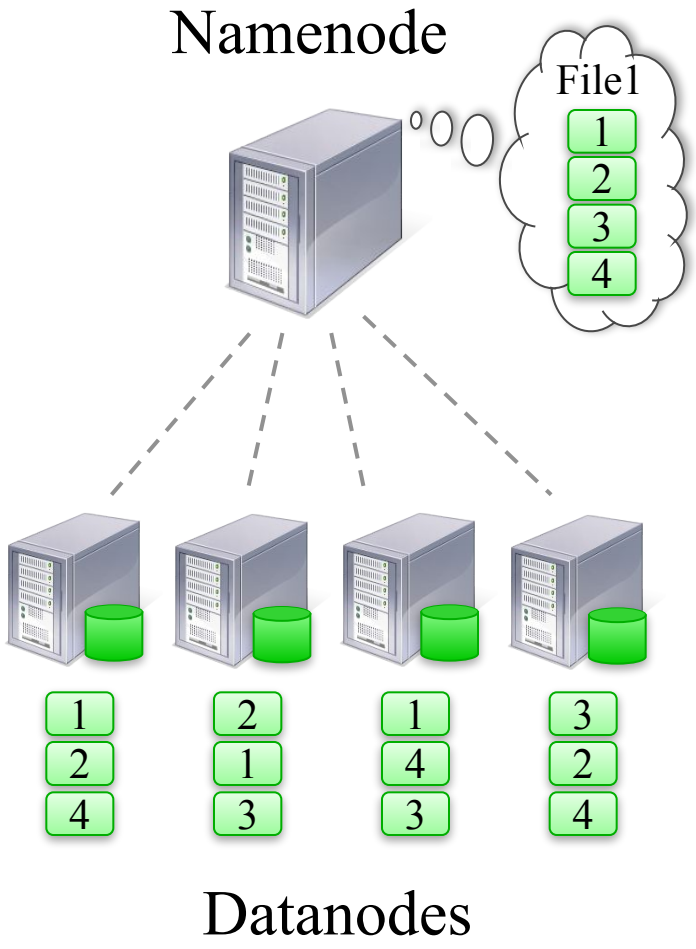


NameNode

- Metadata is stored on a dedicated server, called the NameNode and keeps the entire namespace in RAM, allowing fast access to the metadata.
- Collects Block reports from DataNodes on data block locations.
- Replicates the missing data blocks.
- All nodes are fully connected and communicate with each other using TCP-based protocols (RPC).

NameNode

- Files split into 128MB blocks
- Blocks replicated across several DataNodes (often 3)
- NameNode stores metadata (file names, locations, etc)



300 MB {

A softball-sized eyeball that washed up on the beach is pictured in this October 11, 2012 handout photo from the …

It has been a strange week for things washing up on beaches around the world. On Sunday, a seal was rescued on a beach in Eastbourne, UK, on the coast of the English Channel. The East Sussex Wildlife Rescue and Ambulance Service (WRAS) quickly responded when the animal was reported, and found it with two strange puncture marks and covered in blood.

"We are not sure this stage what has caused the injuries but it may be the rough weather or fighting with another seal." said Trevor Weeks, the founder of WRAS. The seal was otherwise healthy, and after recovering at RSPCA Mallydams, it should be released back into the wild.

The same day, authorities in Cape Town, South Africa, shut down the beaches at the south end of the city after a 40 ton southern right whale washed up on the beach. Several great white sharks were spotted feeding on the whale until it fully washed up on shore.

[Related: Borneo glow-in-the-dark mushrooms rare but do ‘exist outside the psychedelic world’]

"At this stage it is unclear whether the whale was alive when the great white sharks attacked it or whether it died as a result of illness," said disaster management spokesman Wilfred Solomons-Johannes, according to South African newspaper The Star….

128 MB{

128 MB{

44 MB {

A softball-sized eyeball that washed up on the beach is pictured in this October 11, 2012 handout photo from the ...
It has been a strange week for things washing up on beaches around the world. On Sunday, a seal was rescued on a beach in Eastbourne, UK, on the coast of the English Channel. The East Sussex Wildlife Rescue and Ambulance Service (WRAS) quickly responded when the animal was reported, and found it with two strange puncture marks and covered in blood.
"We are not sure this stage what has caused the injuries but it may be the rough weather or fighting with another seal." said Trevor Weeks, the founder of WRAS. The seal was otherwise healthy, and after recovering at RSPCA Mallydams, it should be released back into the wild.
The same day, authorities in Cape Town, South Africa, shut down the beaches at the south end of the city after a 40 ton southern right whale washed up on the beach. Several great white sharks were spotted feeding on the whale until it fully washed up on shore.
[Related: Borneo glow-in-the-dark mushrooms rare but do 'exist outside the psychedelic world']
"At this stage it is unclear whether the whale was alive when the great white sharks attacked it or whether it died as a result of illness," said disaster management spokesman Wilfred Solomons-Johannes, according to South African newspaper The Star....

x3

A softball-sized eyeball that washed up on the beach is pictured in this October 11, 2012 handout photo from the ...
It has been a strange week for things washing up on beaches around the world. On Sunday, a seal was rescued on a beach in Eastbourne, UK, on the coast of the English Channel. The East Sussex Wildlife Rescue and Ambulance Service (WRAS) quickly responded when the animal was reported, and found it with two strange puncture marks and covered

x3

in blood.
"We are not sure this stage what has caused the injuries but it may be the rough weather or fighting with another seal." said Trevor Weeks, the founder of WRAS. The seal was otherwise healthy, and after recovering at RSPCA Mallydams, it should be released back into the wild.
The same day, authorities in Cape Town, South Africa, shut down the beaches at the south end of the city after a 40 ton southern right whale washed up on the beach Several great white sharks were

x3

spotted feeding on the whale until it fully washed up on shore.
[Related: Borneo glow-in-the-dark mushrooms rare but do ‘exist outside the psychedelic world’]
"At this stage it is unclear whether the whale was alive when the great white sharks attacked it or whether it died as a result of illness," said disaster management spokesman Wilfred Solomons-Johannes, according to South African newspaper The Star....

What the Namespace Node Does

- Listens for Heartbeats
- Listens for Client Requests
- If no heartbeat
  - marks a node as dead
  - Its data is deregistered
- It selects DataNodes
  - Which nodes get which blocks
  - Signals creating, opening, closing
- Deletes
  - Orders move to /trash
  - Starts delete timer

NameNode

State is stored in two files:

- **FS image:** Snapshot of file system metadata
- **Edit log:** Changes since last snapshot

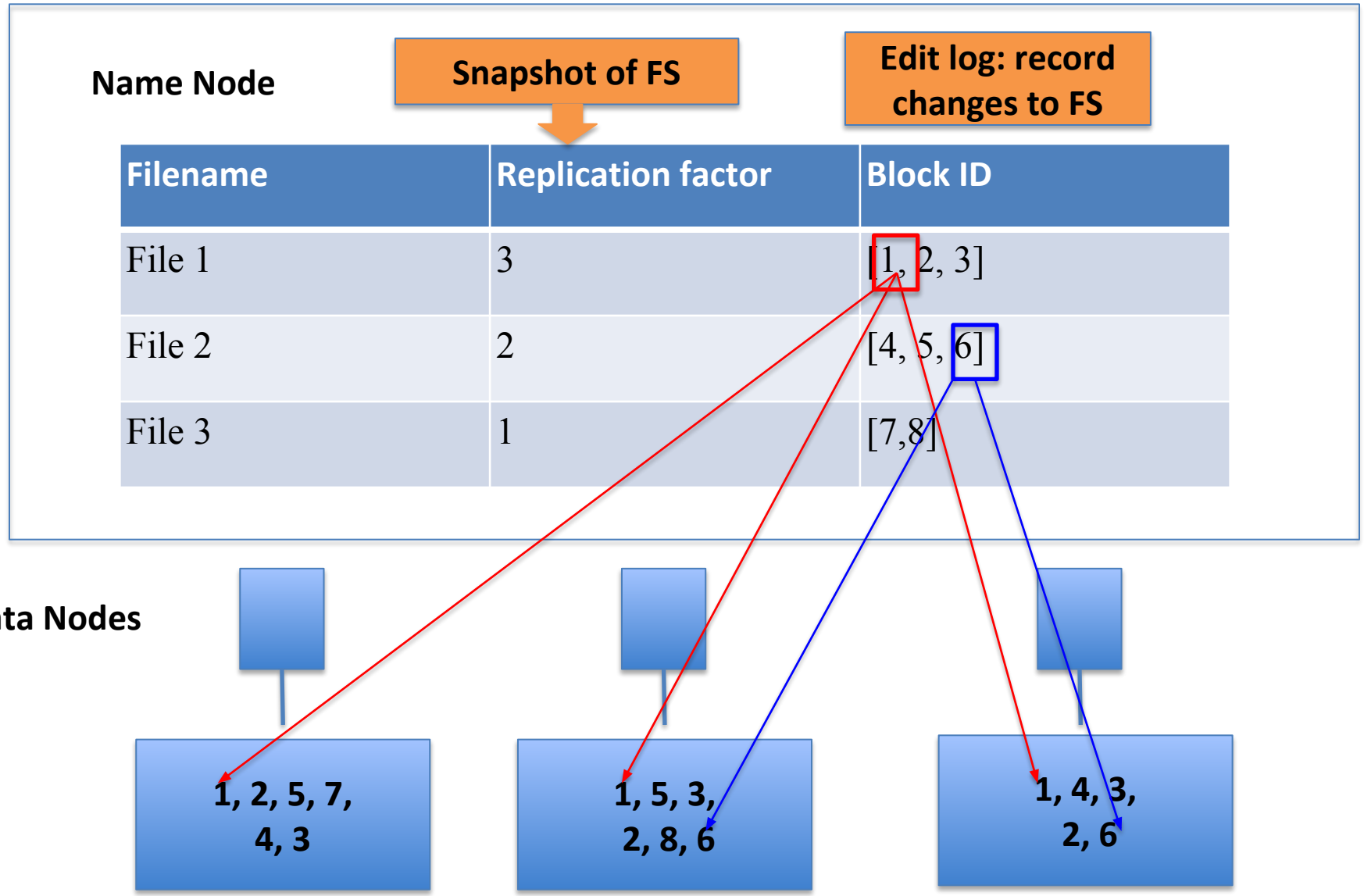
Normal Operation:

When NameNode starts, it reads **FS image** and then applies all the changes from edits sequentially

Snapshot

- Persistently save current state
- Instruction during handshake

HDFS Inside: Name Node



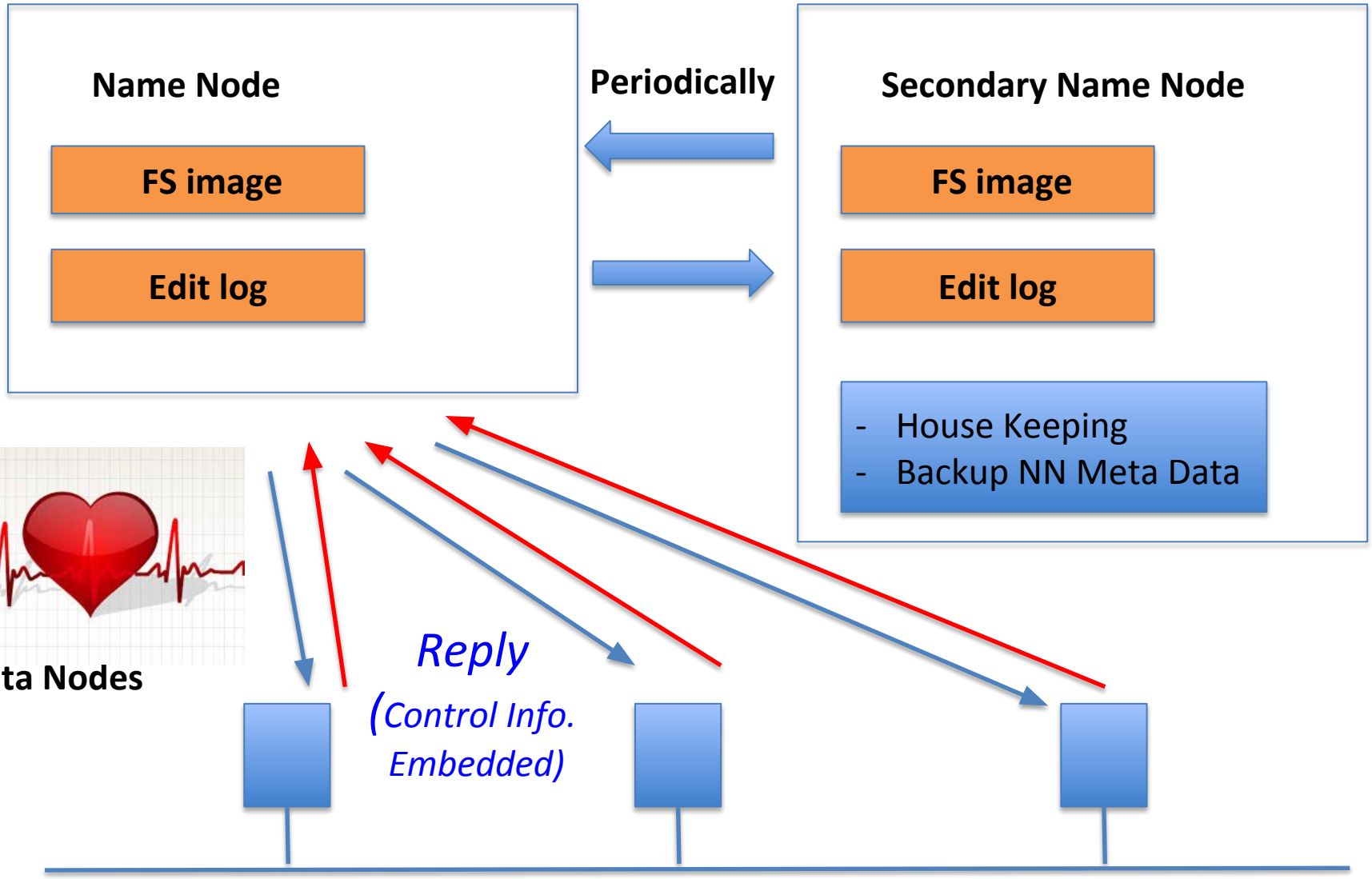
Metadata Disk Failure

- A corruption of these files can cause a HDFS instance to be non-functional.
- For this reason, a NameNode can be configured to maintain multiple copies of the FS image and EditLog.
- Multiple copies of the FS image and EditLog files are updated synchronously.
- Meta-data is not data-intensive.
- The NameNode could be single point failure: automatic failover is NOT

The Secondary NameNode

- Periodically merge namespace image with Edit log
- NOT a backup for the NameNode
- If the NameNode fails, it does not take over the responsibilities of NN
- Periodically reads the log file and applies the changes to the FS image file bringing it up to date
- Runs on separate physical machine
- Has a copy of metadata, which can be used to reconstruct state of the NameNode

HDFS Inside: Name Node



DataNodes

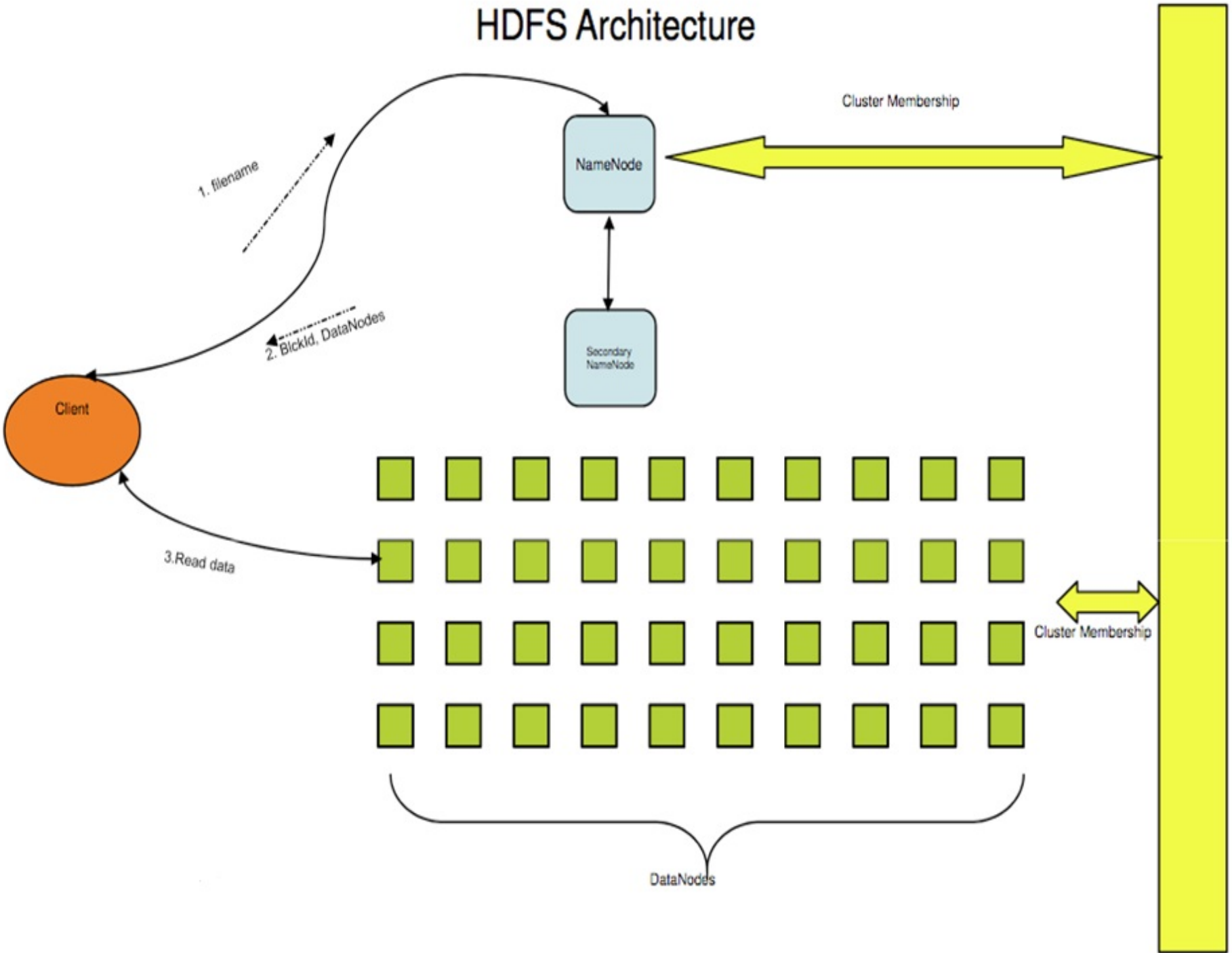
- A file is split into one or more blocks and set of blocks are stored in DataNodes.
- The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.
- DataNode has no knowledge about HDFS filesystem
- DataNode does not create all files in the same directory.
- When the filesystem starts up it generates a list of all HDFS blocks and send this report to NameNode

The Communication Protocol

- All HDFS communication protocols are layered on top of the TCP/IP protocol
- A client establishes a connection to a configurable TCP port on the NameNode. It talks ClientProtocol with the NameNode.
- The DataNodes talk to the NameNode using DataNode protocol.
- RPC abstraction wraps both ClientProtocol and DataNode protocol.
- NameNode never initiates a request; it only responds to RPC requests issued by DataNodes or clients.

Handshake

- During startup each DataNode connects to the NameNode and performs a handshake.
- The purpose of the handshake is to verify the namespace ID and the software version of the DataNode.
- A DataNode that is newly initialized will receive the cluster's namespace ID and permitted to join the cluster.
- Nodes with a different namespace ID will not be able to join the cluster, thus preserving the integrity of the file system.
- After the handshake DataNode registers with the NameNode.


HDFS Architecture
Cluster Membership
NameNode
1. filename
2. BlckId, DataNodes
Secondary NameNode
Client
3.Read data
Cluster Membership
DataNodes

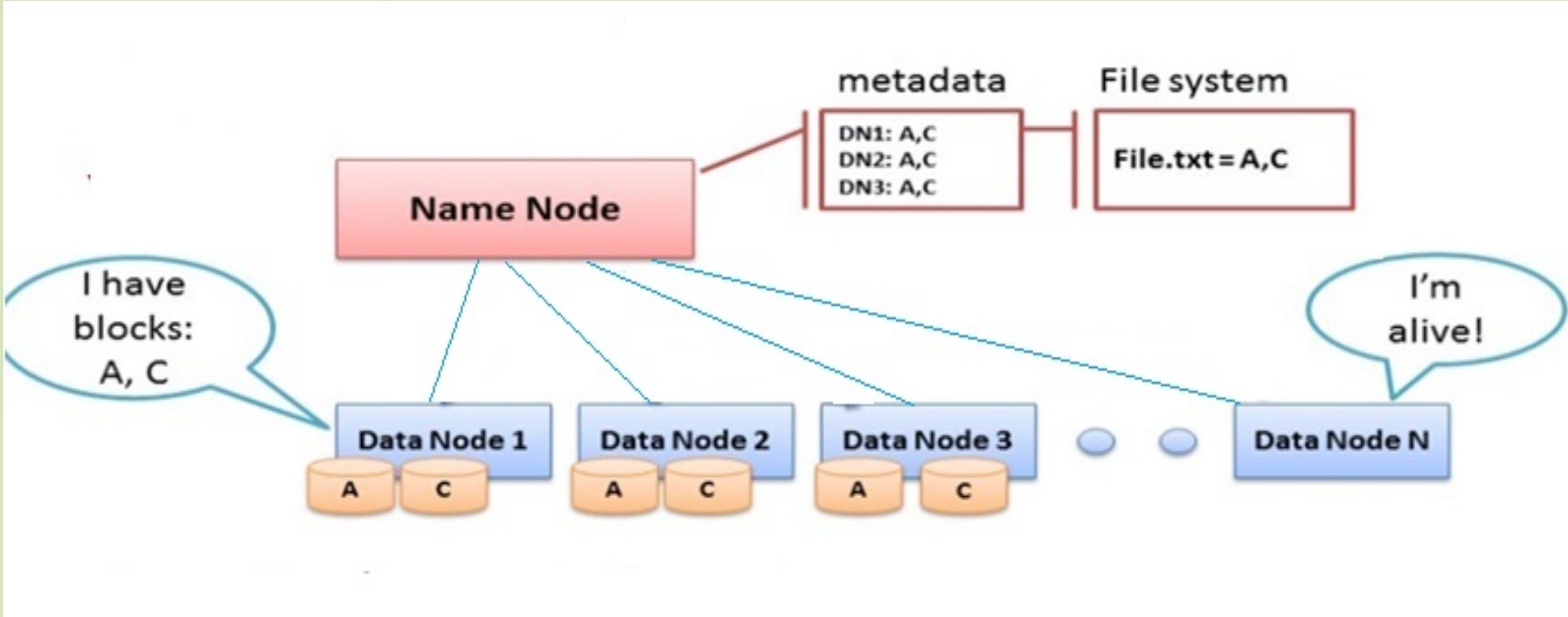
Block Reports

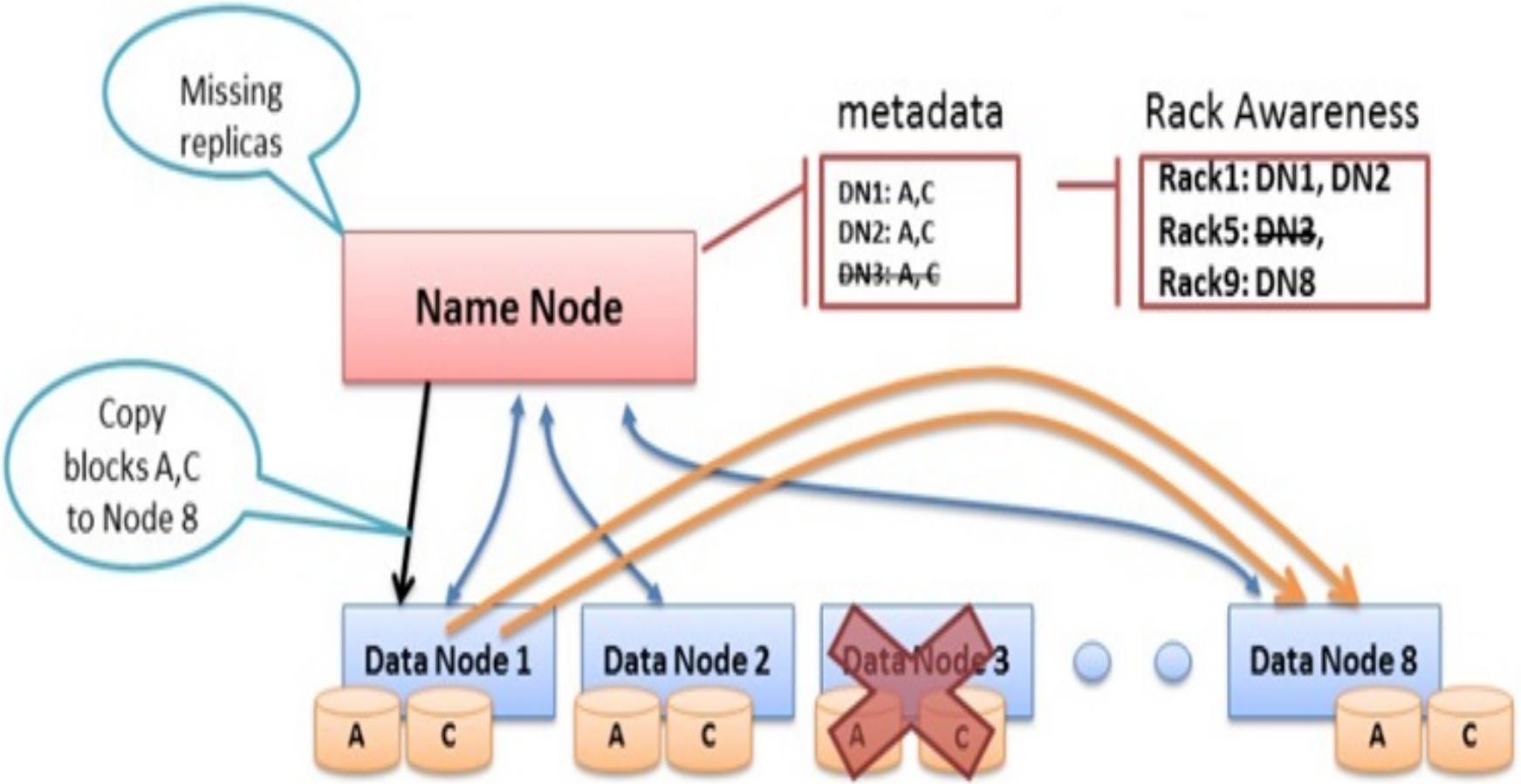
- A DataNode identifies block replicas in its possession and sends block report to the NameNode.
- A block report contains the block id and the generation stamp.
- The first block report is sent immediately after the DataNode registration.
- Subsequent block reports are sent periodically and provide the NameNode with an up-to-date view of where block replicas are located on the cluster.

Heartbeats

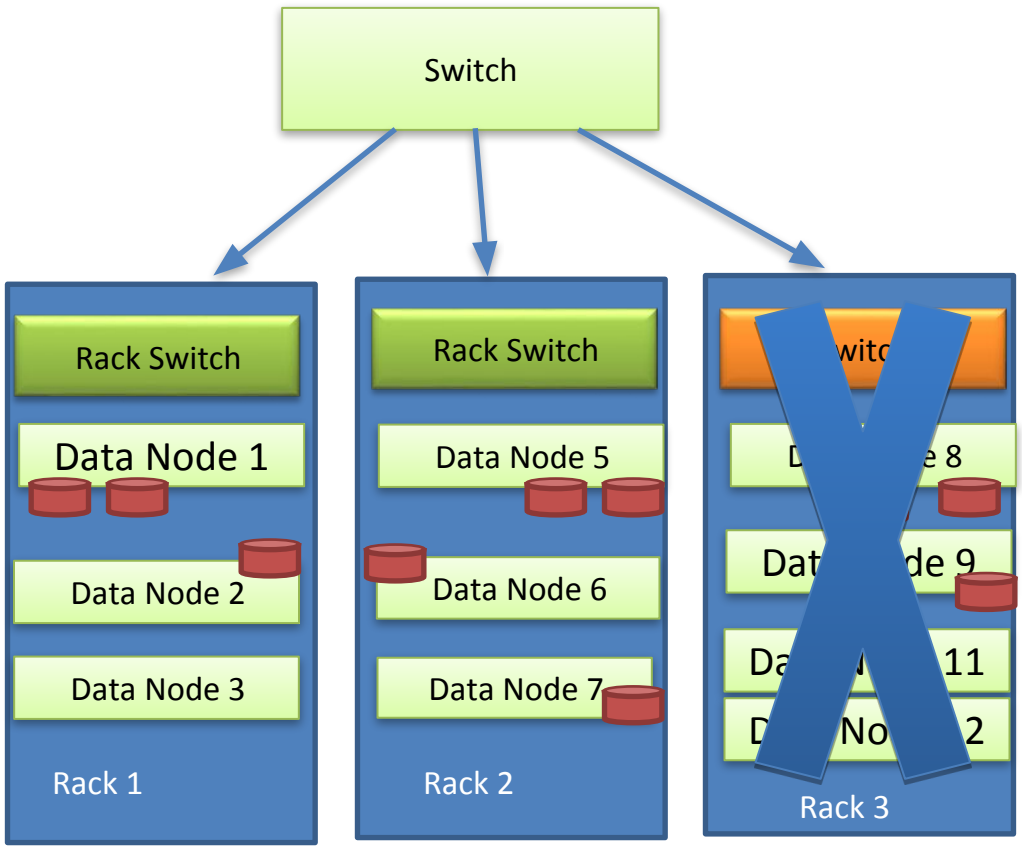
- NameNode and DataNode communication is through Heartbeats.
- During normal operation DataNodes will send heartbeats to the NameNode to confirm that the DataNode is operating and the block replicas it hosts are available.
- The default heartbeat interval is three seconds.
- If the NameNode does not receive a heartbeat from a DataNode in ten minutes then the NameNode considers that the DataNode is out of service.

- The NameNode does not directly calls DataNodes.
- It replies to heartbeats to send instructions to the DataNodes.
- The NameNode can process thousands of heartbeats per second without affecting other NameNode operations.




Missing replicas
metadata
Rack Awareness
DN1: A,C
DN2: A,C
DN3: A, C
Rack1: DN1, DN2
Rack5: DN3,
Rack9: DN8
Name Node
Copy blocks A,C to Node 8
Data Node 1
Data Node 2
Data Node 3
Data Node 8
A
C
A
C
A
C
A
C

Hadoop Rack Awareness



Name Node

Rack Aware

Rack 1:
DataNode 1
DataNode 2
DataNode 3

Rack 2:
DataNode 5
DataNode 6
DataNode 7

Rack 3:
DataNode 8
DataNode 9
DataNode 11
DataNode 12

metadata

Results.txt =
BLK A:
DN1, DN5, DN6

BLK B:
DN7, DN1, DN2

BLK C
DN5, DN8, DN9

Replication Management

- The NameNode detects that a block has become under or over replicated when a block report from a DataNode arrives.
- When a block becomes over replicated, the NameNode chooses a replica to remove.
- When a block becomes under-replicated, it is put in the replication priority queue.

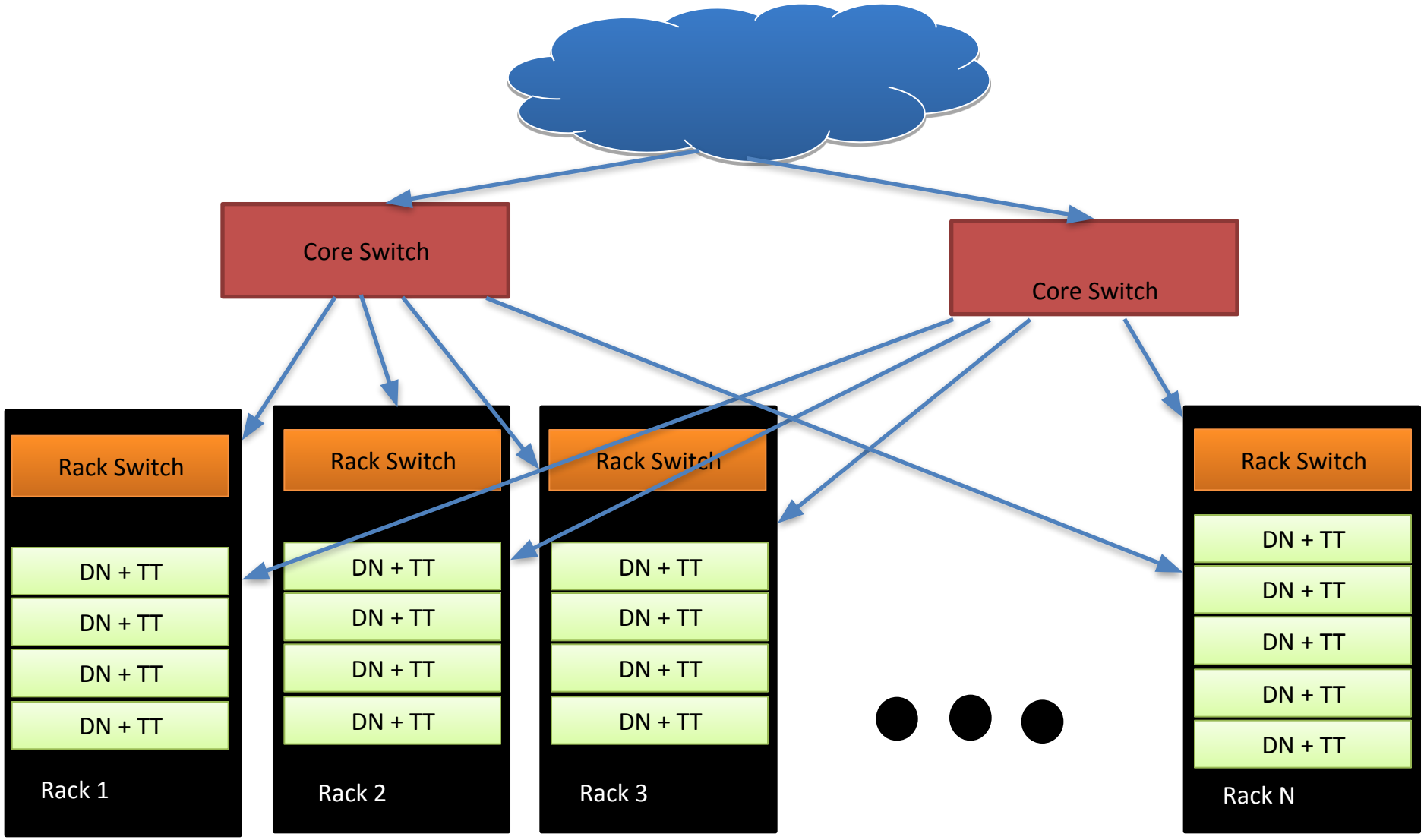
Replica Placement Policy

- The placement of the replicas is critical to HDFS reliability and performance.
- Optimizing replica placement distinguishes HDFS from other distributed file systems.

The default HDFS replica placement policy is as follows:

1. No DataNode contains more than one replica of any block.
2. No rack contains more than two replicas of the same block, provided there are sufficient racks on the cluster.

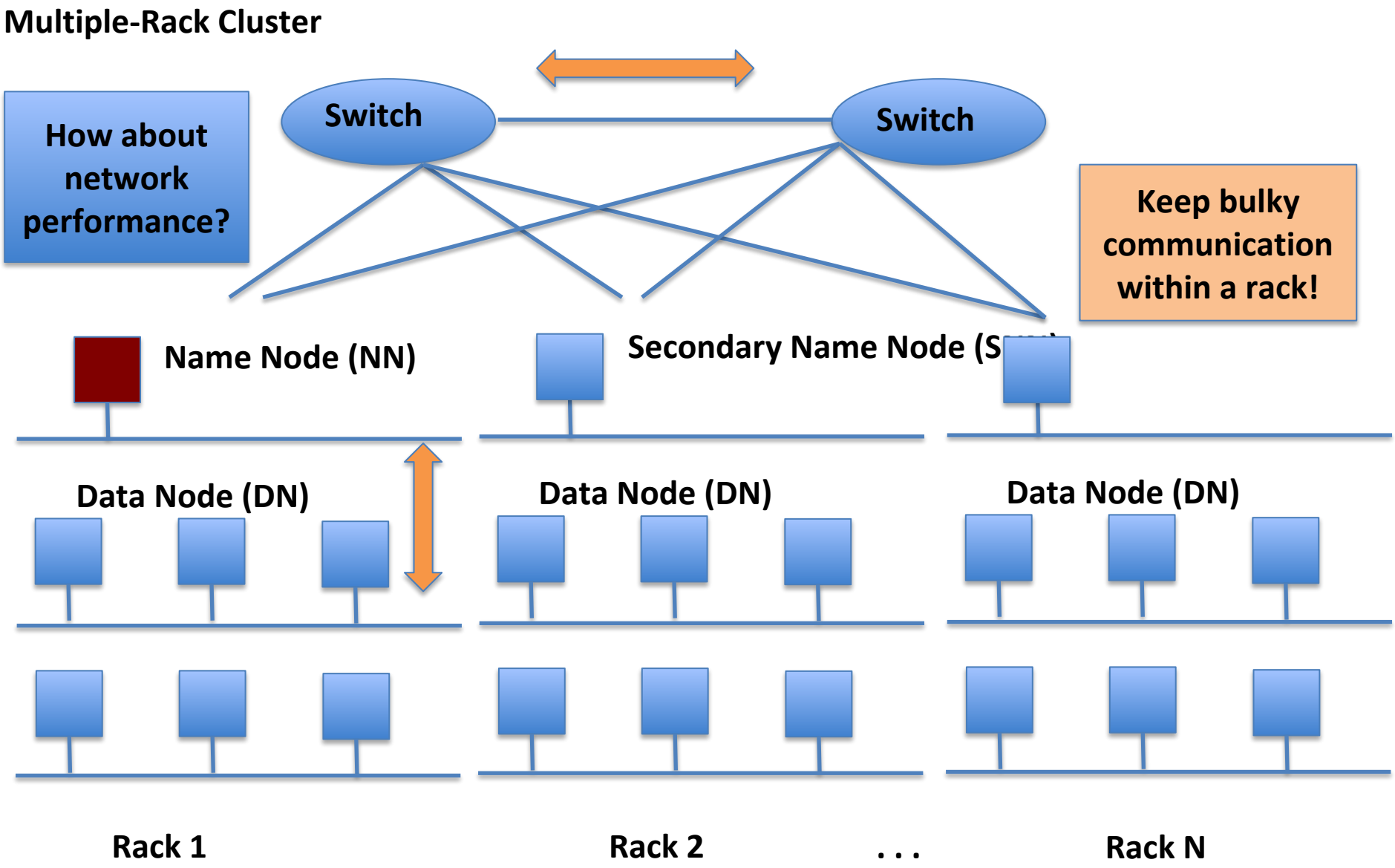
General Hadoop Cluster



Block Placement

- Nodes of a rack share a switch, and rack switches are connected by one or more core switches.
- Communication between two nodes in different racks has to go through multiple switches.
- In most cases, network bandwidth between nodes in the same rack is greater than network bandwidth between nodes in different racks.

HDFS Architecture: Master-Slave

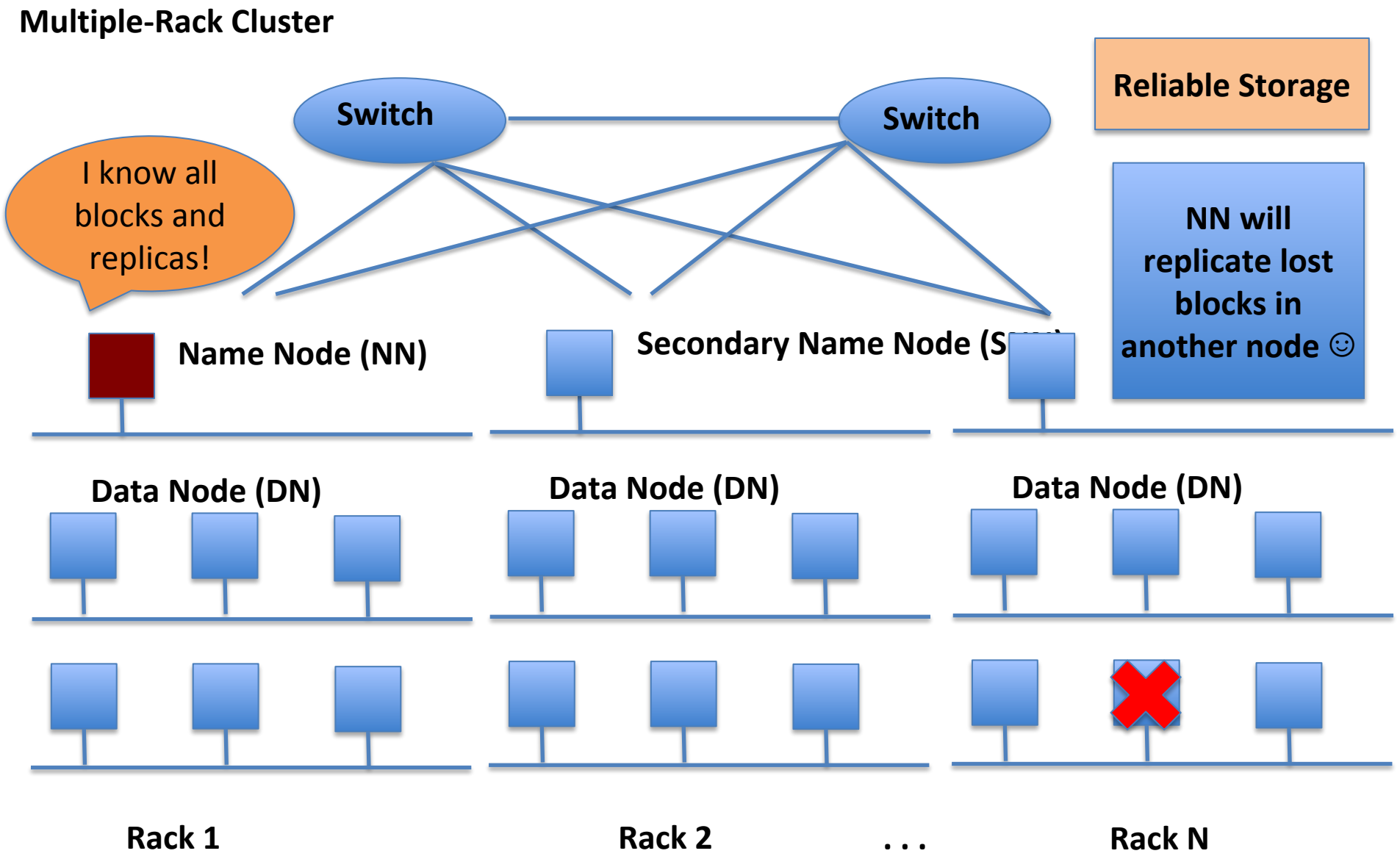


Re-replication

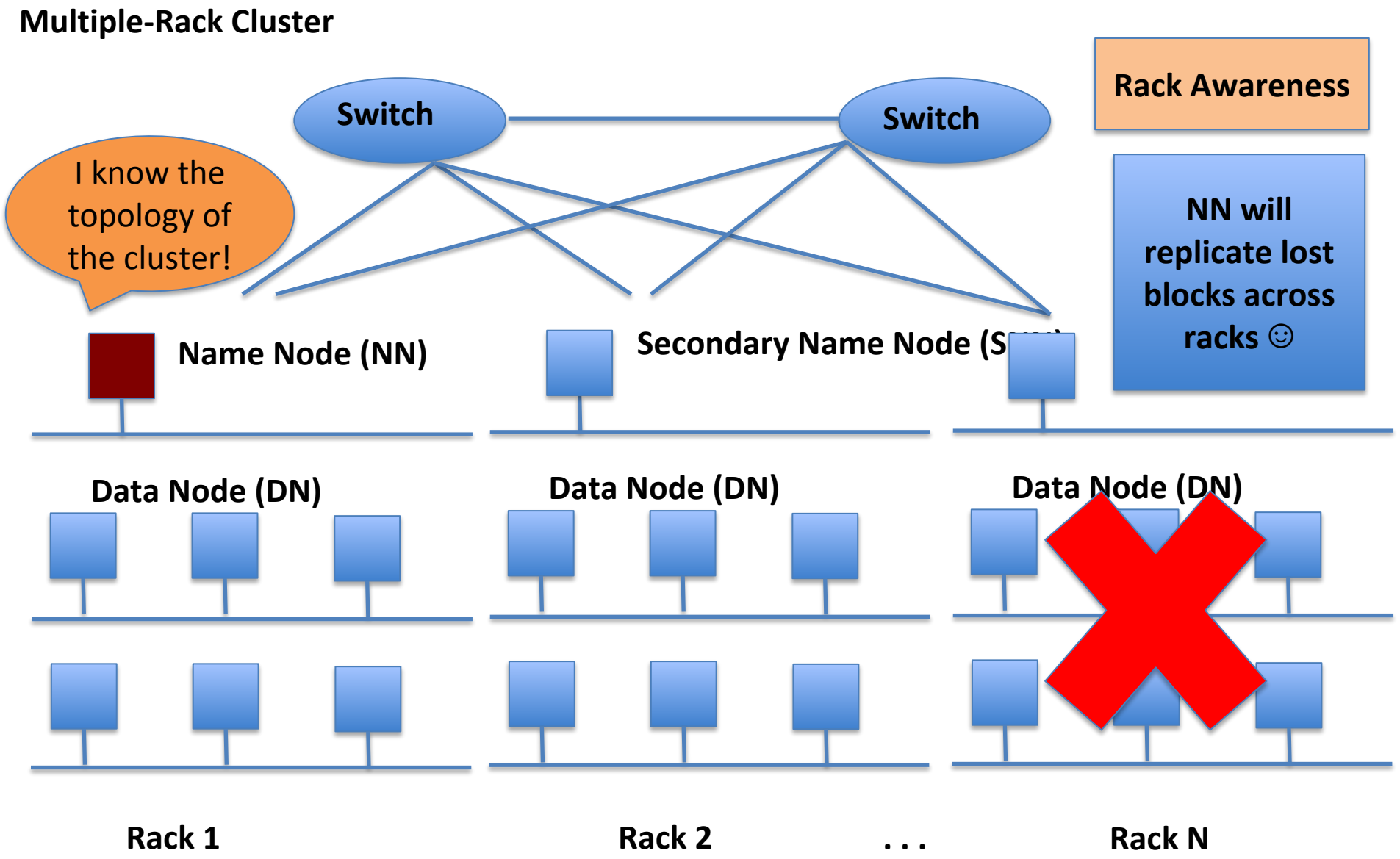
The necessity for re-replication may arise due to:

- A DataNode may become unavailable,
- A replica may become corrupted,
- A hard disk on a DataNode may fail, or
- The replication factor on the block may be increased.

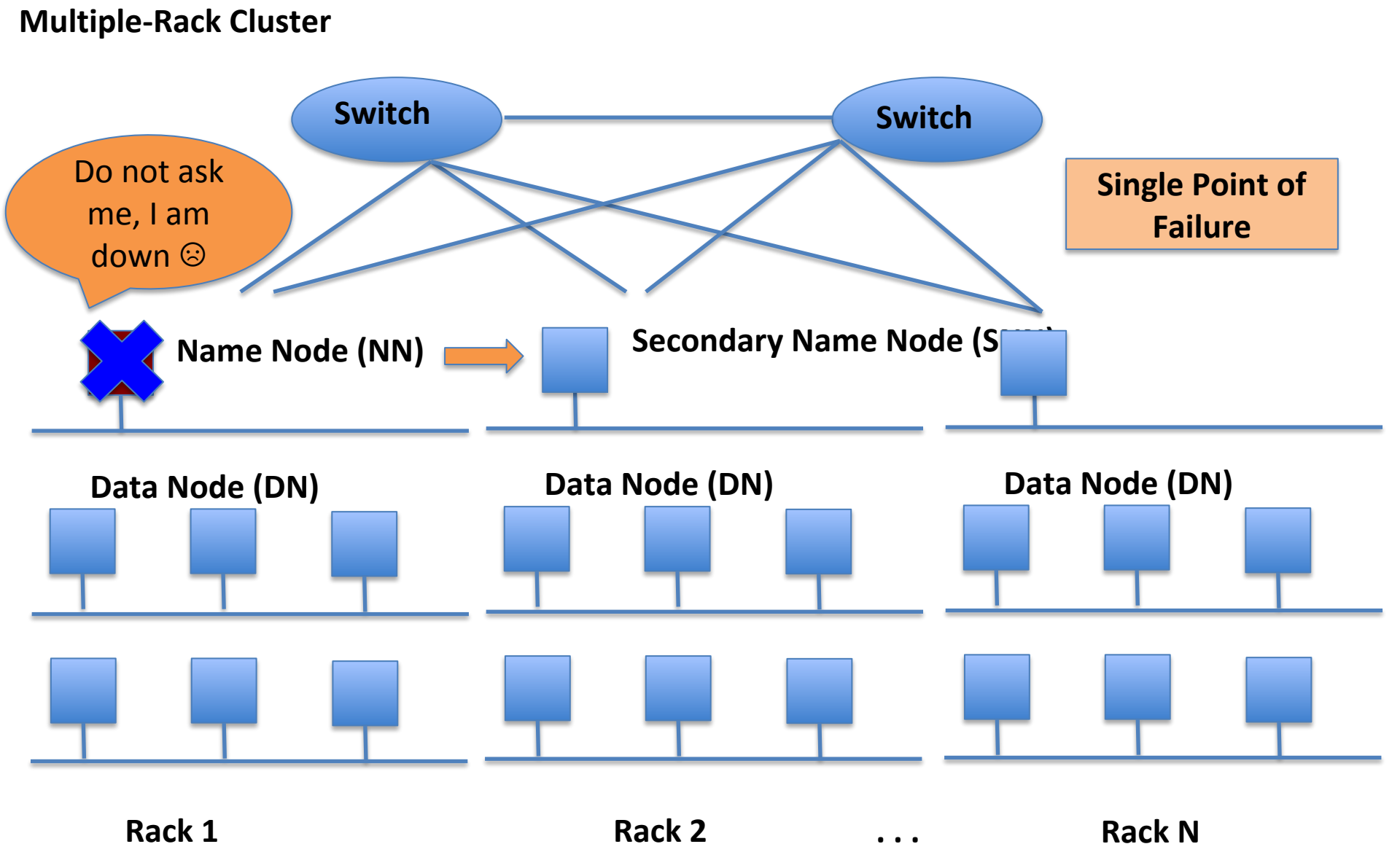
HDFS Architecture: Master-Slave



HDFS Architecture: Master-Slave



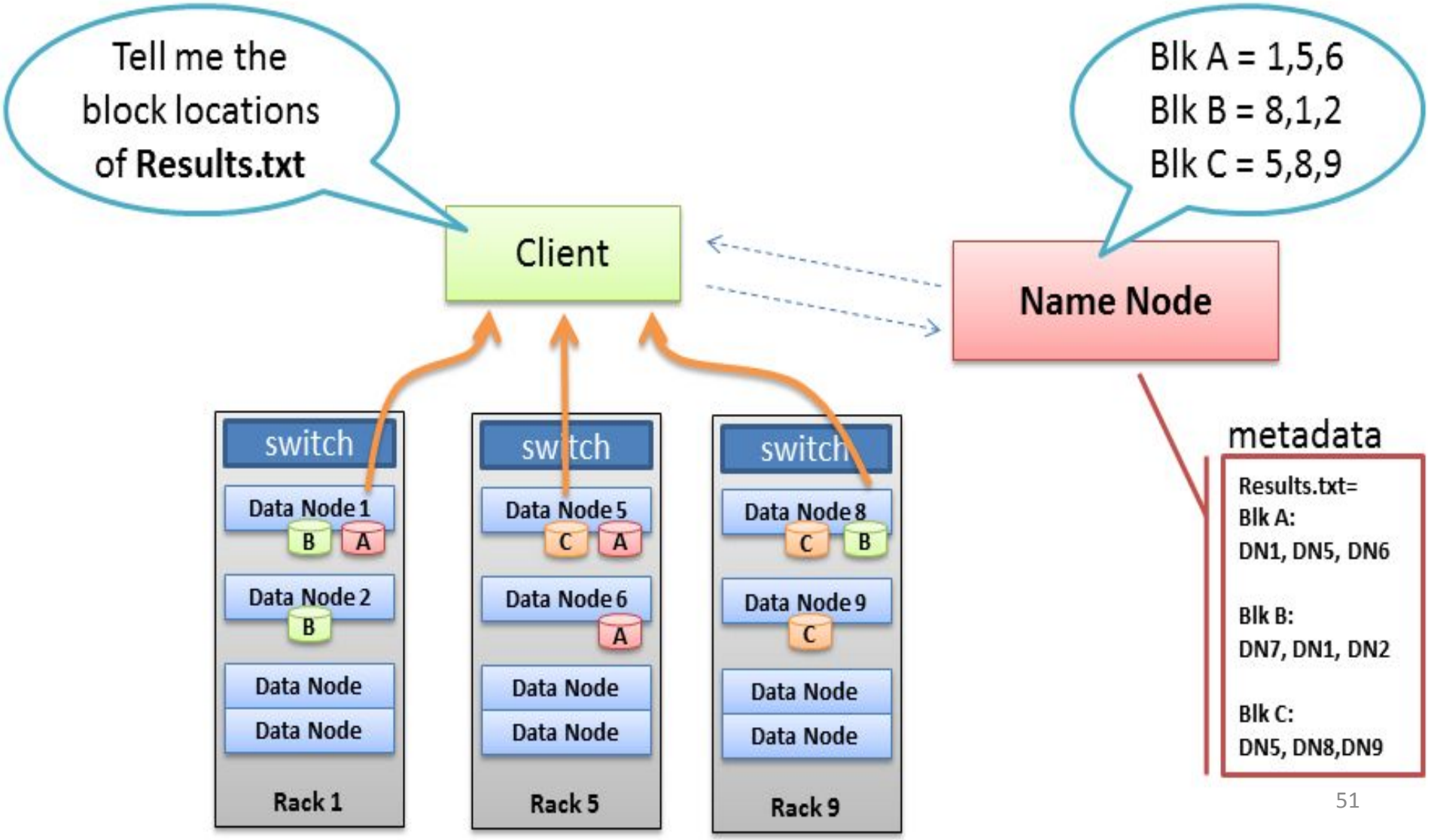
HDFS Architecture: Master-Slave



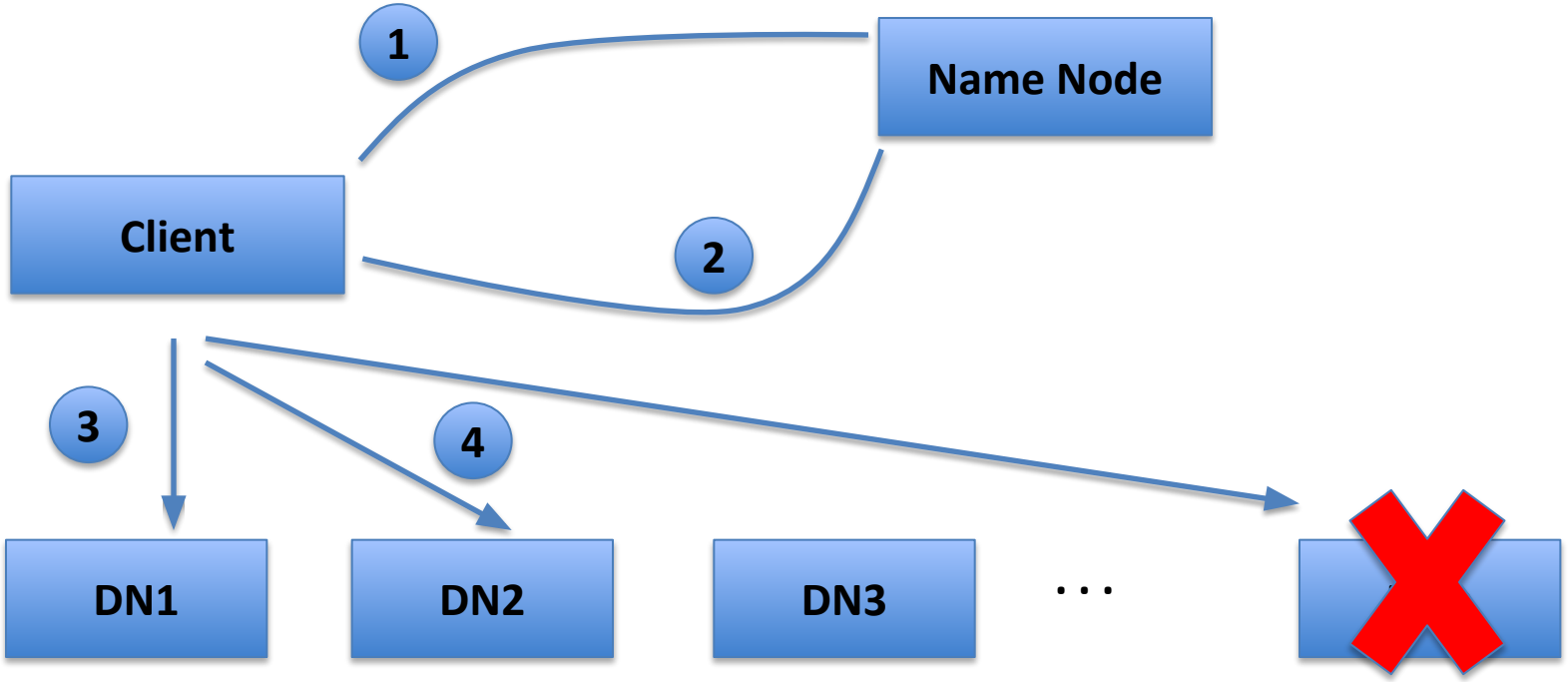
File Read

- User applications access the file system using the HDFS client.
- When an application reads a file, the HDFS client first asks the NameNode for the list of DataNodes that host replicas of the blocks of the file.
- It then contacts a DataNode directly and requests the transfer of the desired block.

Client reading files from HDFS

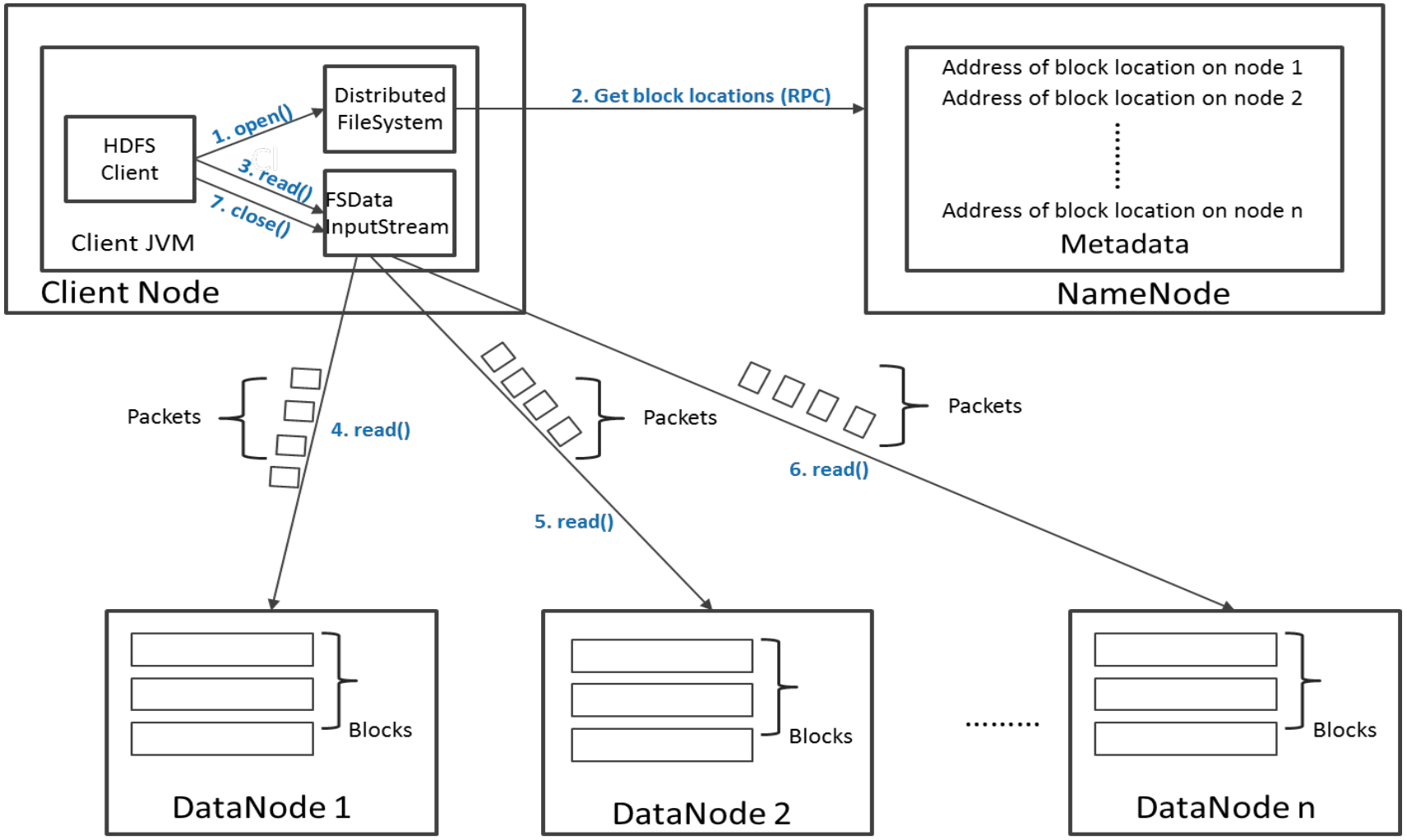


HDFS Inside: Read



1. Client connects to NN to read data
2. NN tells client where to find the data blocks
3. Client reads blocks directly from data nodes (without going through NN)
4. In case of node failures, client connects to another node that serves the missing block

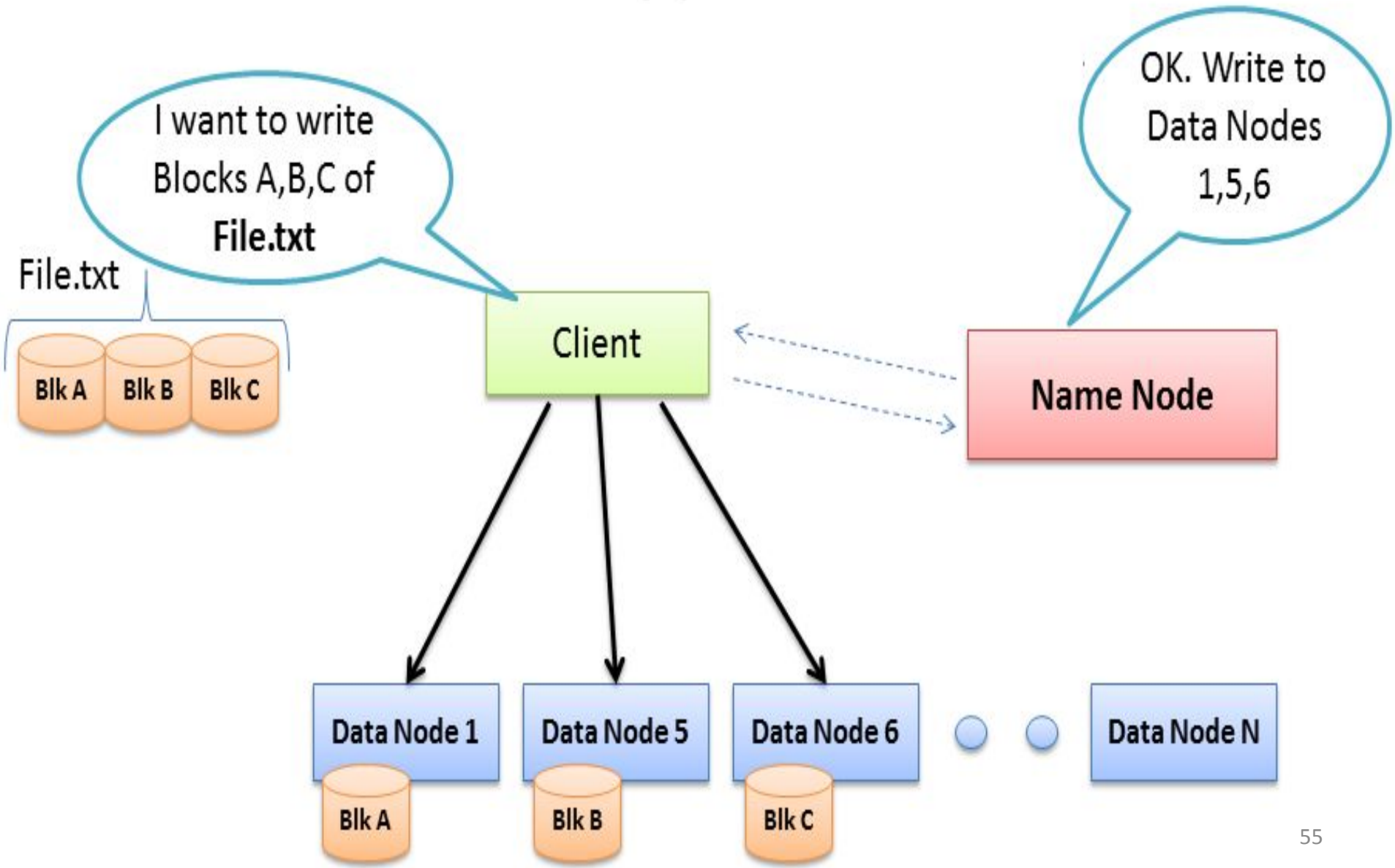
Read Operation in HDFS



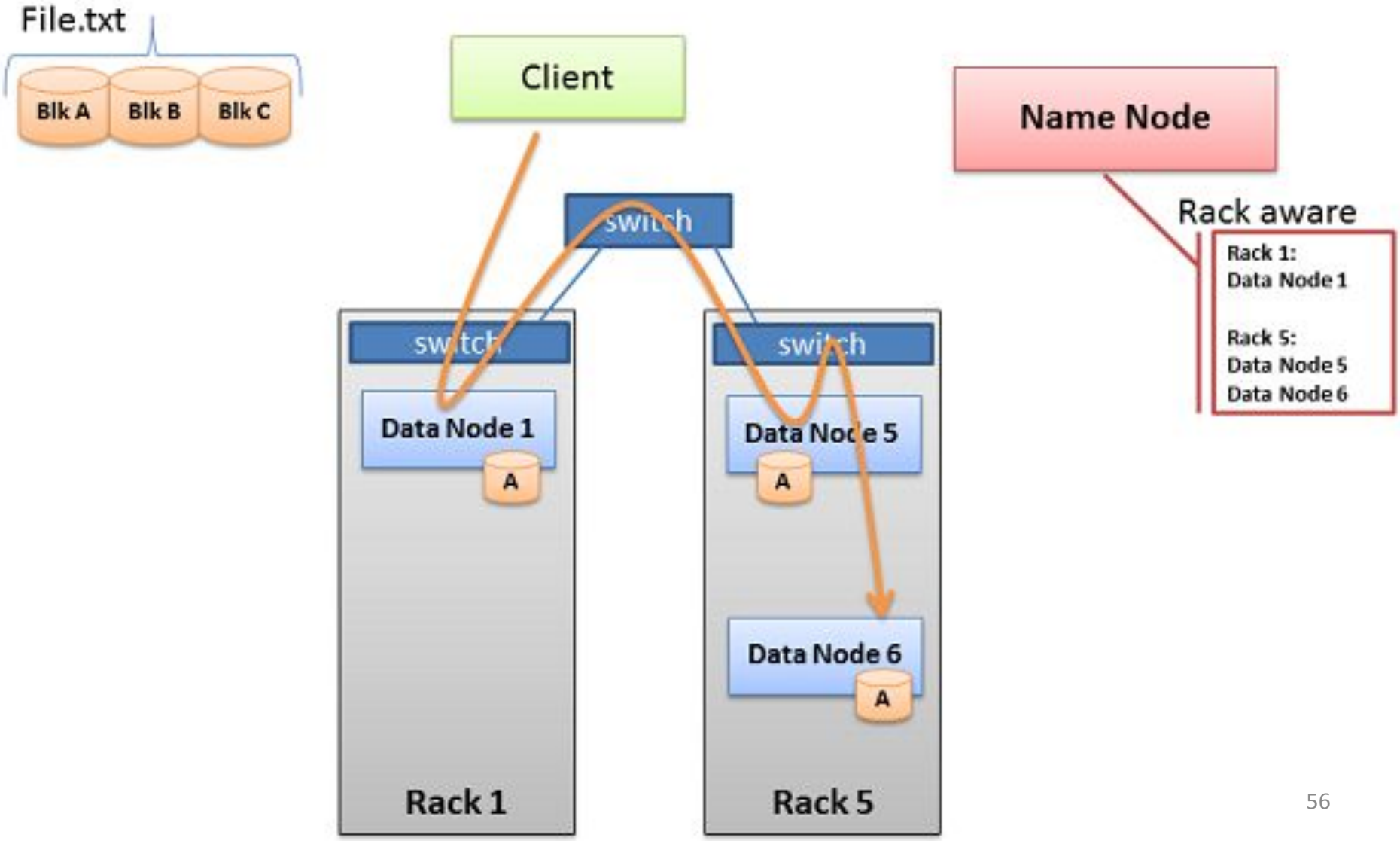
File Write

- When a client writes, it first asks the NameNode to choose DataNodes to host replicas of the first block of the file.
- The client organizes a pipeline from node-to-node and sends the data.
- When the first block is filled, the client requests new DataNodes to be chosen to host replicas of the next block.
- A new pipeline is organized, and the client sends the further bytes of the file.

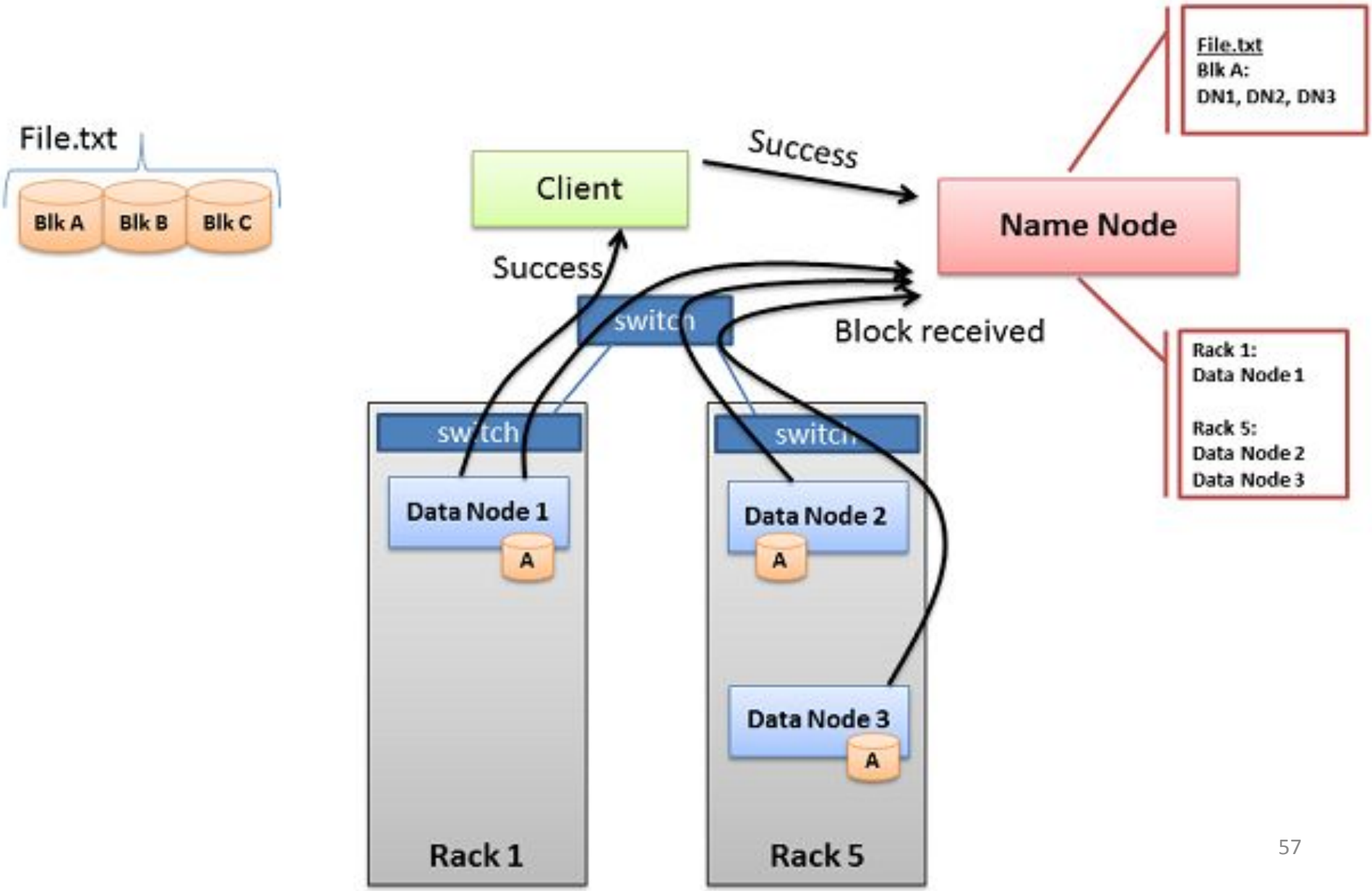
Writing files to HDFS



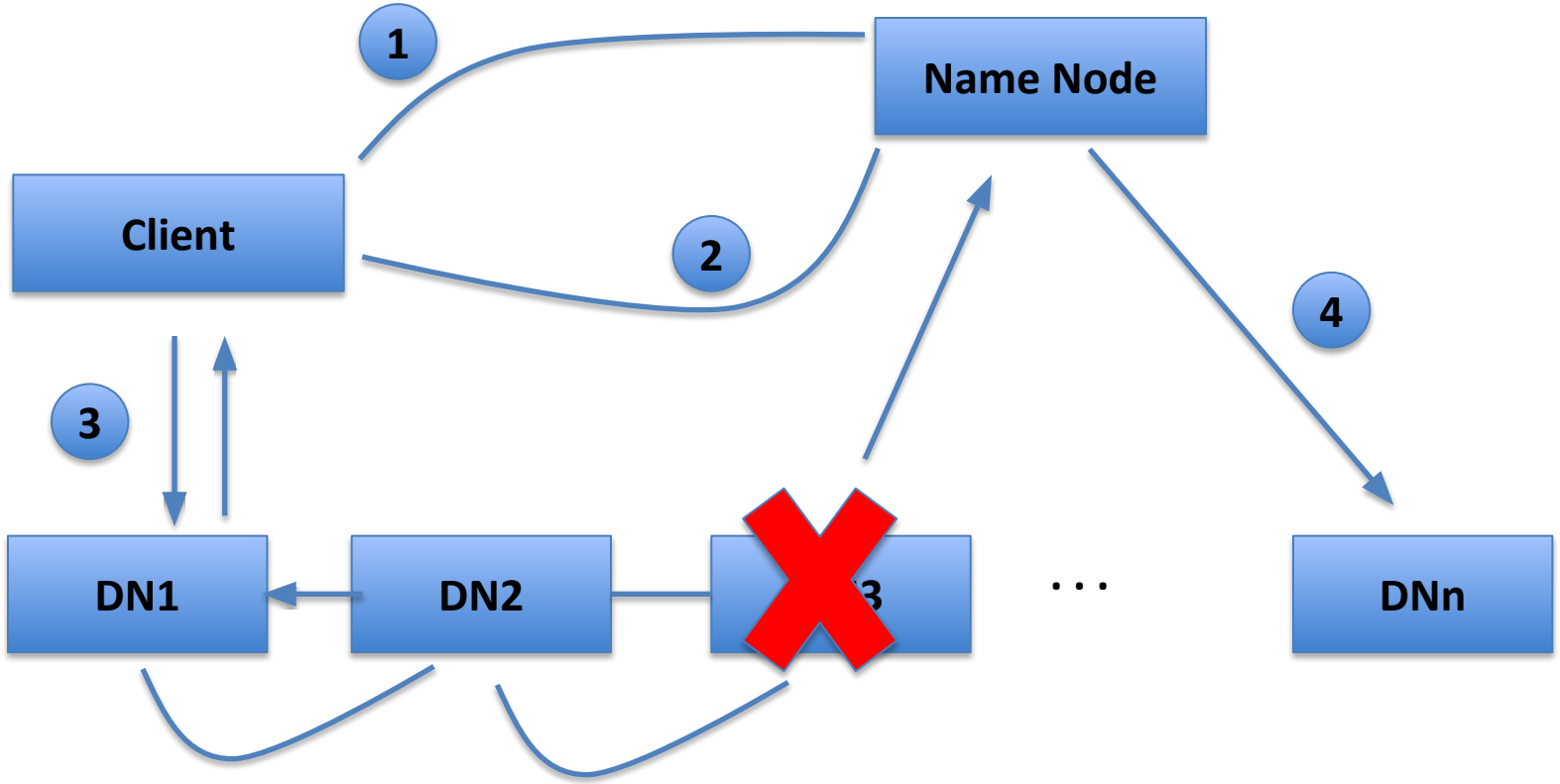
Pipelined write



Pipelined write success

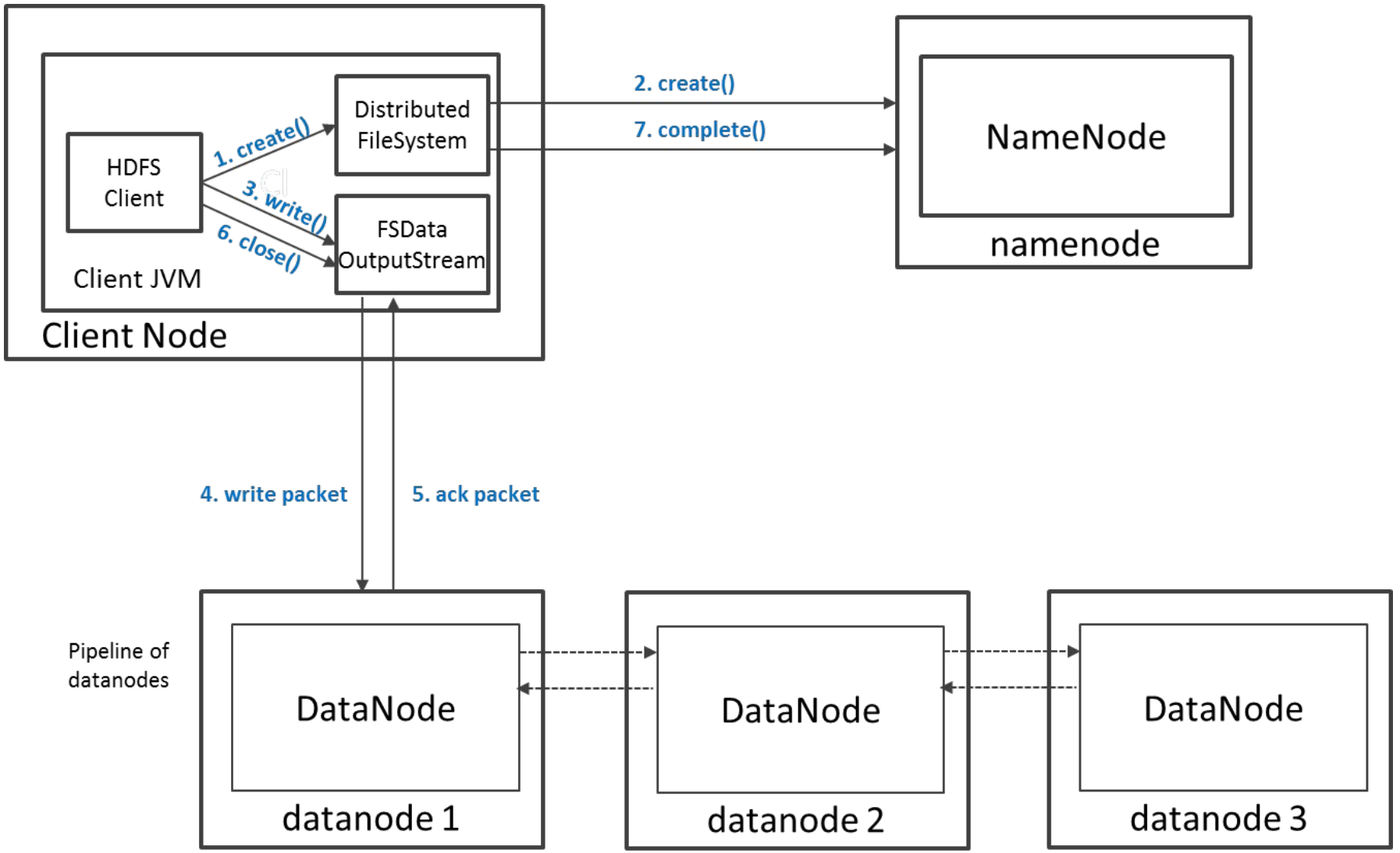


HDFS Inside: Write



1. Client connects to NN to write data
2. NN tells client write these data nodes
3. Client writes blocks directly to data nodes with desired replication factor
4. In case of node failures, NN will figure it out and replicate the missing blocks

Write Operation in HDFS



Cluster Rebalancing

- HDFS architecture is compatible with data rebalancing schemes.
- A scheme might move data from one DataNode to another if the free space on a DataNode falls below a certain threshold.
- In the event of a sudden high demand for a particular file, a scheme might dynamically create additional replicas and rebalance other data in the cluster.

Data Integrity

- Consider a situation: a block of data fetched from DataNode arrives corrupted.
- This corruption may occur because of faults in a storage device, network faults, or buggy software.
- A HDFS client creates the checksum of every block of its file and stores it in hidden files in the HDFS namespace.
- When a clients retrieves the contents of file, it verifies that the corresponding checksums match.
- If does not match, the client can retrieve the block from a replica.

About Data locality